CVXPY x NASA Course 2024

Philipp Schiele Steven Diamond Parth Nobel Akshay Agrawal

July 8, 2024

Regression and Statistical Estimation

Outline

Homework Review

Penalty function approximation

Regression

Model Fitting

Maximum likelihood estimation

Sensitivity Analysis

- Let $p^\star(u, v)$ represent the optimal value of the following perturbed optimization problem

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq u_i, \quad i = 1, \ldots, m \\ & h_i(x) \leq v_i, \quad i = 1, \ldots, p \end{array}$$

- CVXPY computes the dual variables $\lambda^\star \in \mathbf{R}^m$, $\nu^\star \in \mathbf{R}^p$
- If $p^\star(u, v)$ is differentiable at $u = 0, v = 0$, then

$$\lambda^\star = -\frac{\mathrm{d}p^\star(0,0)}{\mathrm{d}u}, \quad \nu^\star = -\frac{\mathrm{d}p^\star(0,0)}{\mathrm{d}v}(0).$$

- More generally,

$$p^\star(u, v) \geq p^\star(0, 0) - \lambda^{\star T} u - \nu^{\star T} v.$$

- *Convex Optimization*, §5.6.

Sensitivity analysis homework

- Consider the convex optimization problem

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_1(x) \leq s \\ & Ax = b \end{array}$$

- variable $x \in \mathbf{R}^n$, parametrized by $s \in \mathbf{R}$
- $\lambda^\star$ the optimal dual variable for the inequality constraint with $s = s_{\text{nom}}$.

Sensitivity analysis homework

If $\lambda^\star$ is large, then decreasing s below s_{nom}

- might decrease the optimal value
- will increase the optimal value a lot
- can leave the optimal value unchanged

Sensitivity analysis homework

If $\lambda^\star$ is large, then increasing s above s_{nom}

- will decrease the optimal value a lot
- will increase the optimal value a lot
- can leave the optimal value unchanged

Sensitivity analysis homework

If $\lambda^\star = 0$, then increasing s above s_{nom}

- can decrease the objective value
- can increase the objective value
- will leave the optimal value unchanged

Robust Kalman filtering

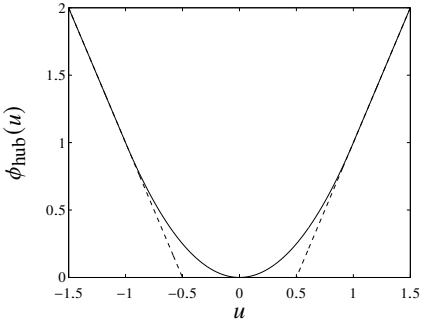
- the Kalman filter works well when v_t and w_t are Gaussian
- we can use a robust optimization approach to make the filter more robust to outliers
- using the Huber penalty function, we can solve

$$\begin{array}{ll} \text{minimize} & \sum_{t=0}^{N-1} \phi_\epsilon(w_t) + \tau\phi_\eta(v_t) \\ \text{subject to} & x_{t+1} = Ax_t + Bw_t \\ & y_t = Cx_t + v_t, \end{array}$$

- where ϕ_ϵ encodes our understanding for the noise in the state transitions
- where ϕ_η encodes our understanding for the noise in the observations

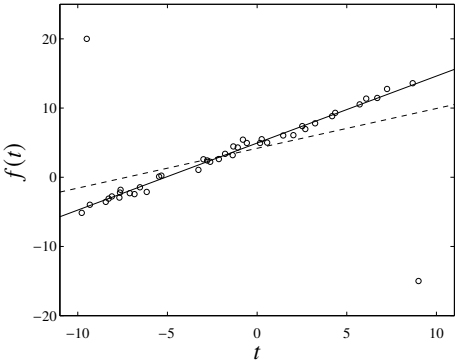
Huber penalty function

$$\phi_{\text{hub}}(u) = \begin{cases} u^2 & |u| \leq M \\ M(2|u| - M) & |u| > M \end{cases}$$



- linear growth for large u makes approximation less sensitive to outliers
- called a **robust penalty**

Example



- 42 points (circles) t_i, y_i, with two outliers
- affine function $f(t) = \alpha + \beta t$ fit using quadratic (dashed) and Huber (solid) penalty

Notebook solutions

- https://marimo.app/l/fqixko

Outline

Homework Review

Penalty function approximation

Regression

Model Fitting

Maximum likelihood estimation

Penalty function approximation

$$\begin{array}{ll} \text{minimize} & \phi(r_1) + \cdots + \phi(r_m) \\ \text{subject to} & r = Ax - b \end{array}$$

($A \in \mathbf{R}^{m \times n}$, $\phi : \mathbf{R} \to \mathbf{R}$ is a convex penalty function)

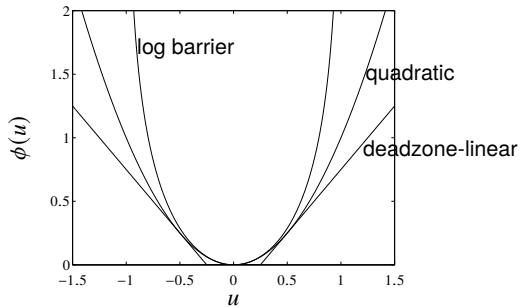
examples

- quadratic: $\phi(u) = u^2$
- deadzone-linear with width a:

$$\phi(u) = \max\{0, |u| - a\}$$

- log-barrier with limit a:

$$\phi(u) = \begin{cases} -a^2 \log(1 - (u/a)^2) & |u| < a \\ \infty & \text{otherwise} \end{cases}$$



Example: histograms of residuals

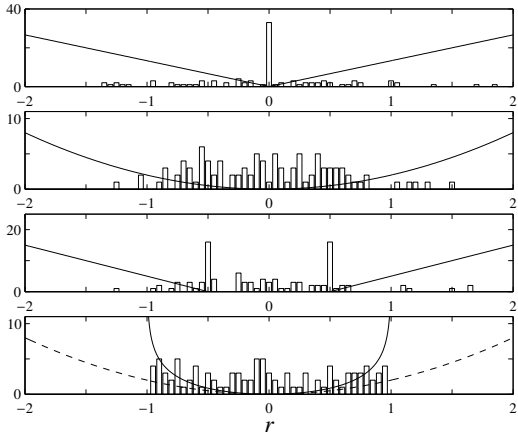
$A \in \mathbf{R}^{100\times 30}$; shape of penalty function affects distribution of residuals

absolute value $\phi(u) = |u|$

square $\phi(u) = u^2$

deadzone $\phi(u) = \max\{0, |u| - 0.5\}$

log-barrier $\phi(u) = -\log(1 - u^2)$



Outline

Homework Review

Penalty function approximation

Regression

Model Fitting

Maximum likelihood estimation

Standard regression

- given data $(x_i, y_i) \in \mathbf{R}^n \times \mathbf{R}$, $i = 1, \ldots, m$
- fit linear (affine) model $\hat{y}_i = \beta^T x_i - v$, $\beta \in \mathbf{R}^n$, $v \in \mathbf{R}$
- residuals are $r_i = \hat{y}_i - y_i$
- least-squares: choose β, v to minimize $\|r\|_2^2 = \sum_i r_i^2$
- mean of optimal residuals is zero
- can add (Tychonov) regularization: with $\lambda > 0$,

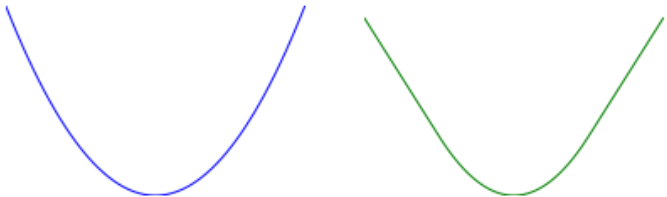
$$\text{minimize} \quad \|r\|_2^2 + \lambda \|\beta\|_2^2$$

Robust (Huber) regression

- replace square with *Huber function*

$$\phi(u) = \begin{cases} u^2 & |u| \leq M \\ 2Mu - M^2 & |u| > M \end{cases}$$

$M > 0$ is the Huber threshold



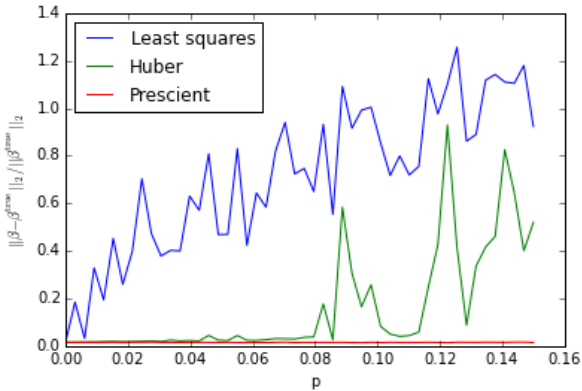
- same as least-squares for small residuals, but allows (some) large residuals

Example

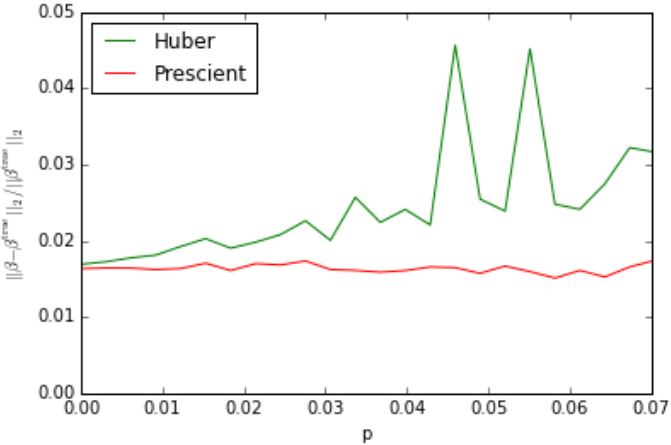
- $m = 450$ measurements, $n = 300$ regressors
- choose β^{true}; $x_i \sim \mathcal{N}(0, I)$
- set $y_i = (\beta^{\text{true}})^T x_i + \epsilon_i$, $\epsilon_i \sim \mathcal{N}(0, 1)$
- with probability p, replace y_i with $-y_i$
- data has fraction p of (non-obvious) wrong measurements
- distribution of 'good' and 'bad' y_i are the same
- try to recover $\beta^{\text{true}} \in \mathbf{R}^n$ from measurements $y \in \mathbf{R}^m$
- 'prescient' version: we know which measurements are wrong

Example

50 problem instances, p varying from 0 to 0.15



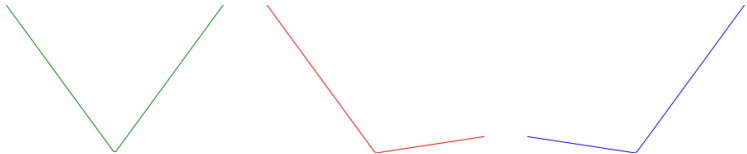
Example


Huber
Prescient
0.05
0.04
0.03
0.02
0.01
0.00
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0.07
p

Quantile regression

- *tilted ℓ_1 penalty*: for $\tau \in (0, 1)$,

$$\phi(u) = \tau(u)_+ + (1-\tau)(u)_- = (1/2)|u| + (\tau - 1/2)u$$



- *quantile regression*: choose β, v to minimize $\sum_i \phi(r_i)$
- $\tau = 0.5$: equal penalty for over- and under-estimating
- $\tau = 0.1$: $9\times$ more penalty for under-estimating
- $\tau = 0.9$: $9\times$ more penalty for over-estimating

Quantile regression

- for $r_i \neq 0$,

$$\frac{\partial \sum_i \phi(r_i)}{\partial v} = \tau \left|\{i : r_i > 0\}\right| - (1-\tau) \left|\{i : r_i < 0\}\right|$$

- (roughly speaking) for optimal v we have

$$\tau \left|\{i : r_i > 0\}\right| = (1-\tau) \left|\{i : r_i < 0\}\right|$$

- and so for optimal v, $\tau m = |\{i : r_i < 0\}|$
- τ-quantile of optimal residuals is zero
- hence the name quantile regression

Example

- time series x_t, $t = 0, 1, 2, \ldots$
- auto-regressive predictor:

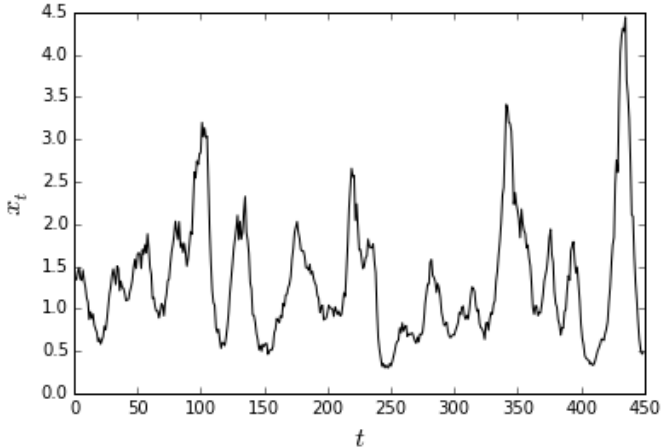
$$\hat{x}_{t+1} = \beta^T (x_t, \ldots, x_{t-M}) - v$$

- $M = 10$ is memory of predictor
- use quantile regression for $\tau = 0.1, 0.5, 0.9$
- at each time t, gives three one-step-ahead predictions:

$$\hat{x}_{t+1}^{0.1}, \qquad \hat{x}_{t+1}^{0.5}, \qquad \hat{x}_{t+1}^{0.9}$$

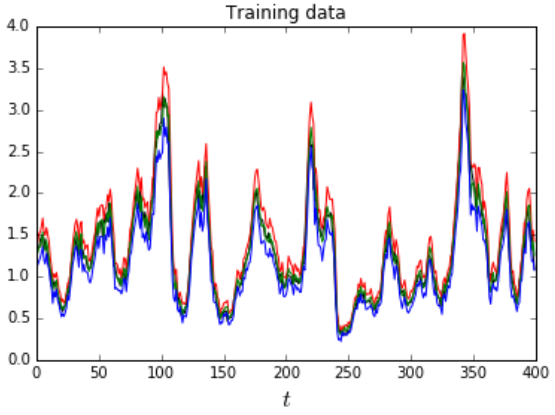
Example

time series x_t



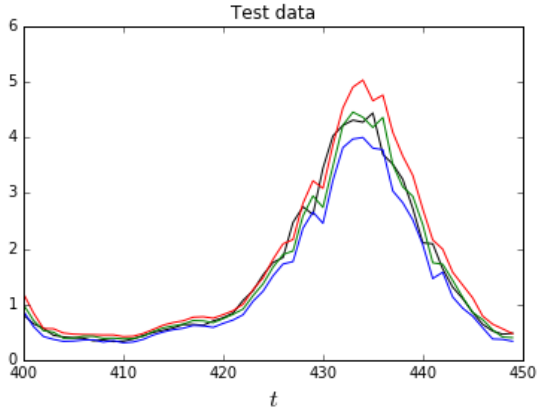
Example

x_t and predictions $\hat{x}_{t+1}^{0.1}, \hat{x}_{t+1}^{0.5}, \hat{x}_{t+1}^{0.9}$ (training set, $t = 0, \ldots, 399$)



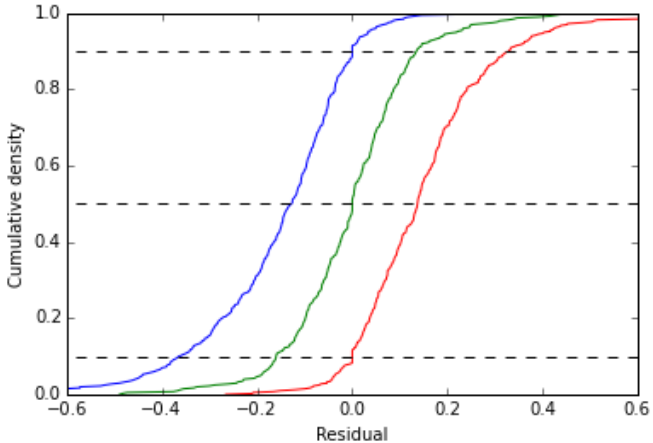
Example

x_t and predictions $\hat{x}_{t+1}^{0.1}, \hat{x}_{t+1}^{0.5}, \hat{x}_{t+1}^{0.9}$ (test set, $t = 400, \ldots, 449$)



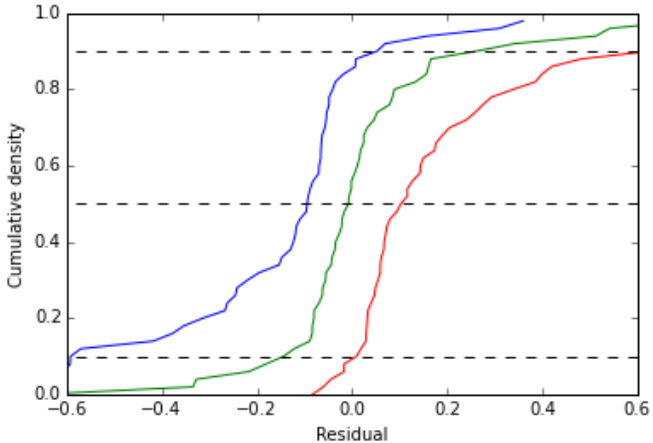
Example

residual distributions for $\tau = 0.9, 0.5,$ and 0.1 (training set)



Example

residual distributions for $\tau = 0.9, 0.5,$ and 0.1 (test set)



Outline

Homework Review

Penalty function approximation

Regression

Model Fitting

Maximum likelihood estimation

Data model

- given data $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, $i = 1, \ldots, m$
- for $\mathcal{X} = \mathbf{R}^n$, x is *feature vector*
- for $\mathcal{Y} = \mathbf{R}$, y is (real) *outcome* or *label*
- for $\mathcal{Y} = \{-1, 1\}$, y is (boolean) outcome

- find *model* or *predictor* $\psi : \mathcal{X} \to \mathcal{Y}$ so that $\psi(x) \approx y$ *for data* (x, y) *that you haven't seen*
- for $\mathcal{Y} = \mathbf{R}$, ψ is a *regression model*
- for $\mathcal{Y} = \{-1, 1\}$, ψ is a *classifier*
- we choose ψ based on observed data, prior knowledge

Loss minimization model

- data model parametrized by $\theta \in \mathbf{R}^n$
- *loss function* $L : \mathcal{X} \times \mathcal{Y} \times \mathbf{R}^n \to \mathbf{R}$
- $L(x_i, y_i, \theta)$ is loss (miss-fit) for data point (x_i, y_i), using model parameter θ
- choose θ; then model is

$$\psi(x) = \operatorname*{argmin}_y L(x, y, \theta)$$

Model fitting via regularized loss minimization

- *regularization* $r : \mathbf{R}^n \to \mathbf{R} \cup \{\infty\}$
- $r(\theta)$ measures model complexity, enforces constraints, or represents prior
- choose θ by minimizing *regularized loss*

$$(1/m) \sum_i L(x_i, y_i, \theta) + r(\theta)$$

- for many useful cases, this is a convex problem
- model is $\psi(x) = \operatorname{argmin}_y L(x, y, \theta)$

Examples

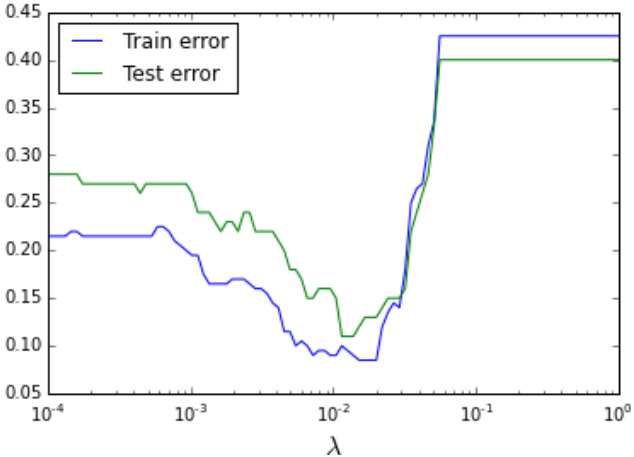
model	$L(x, y, \theta)$	$\psi(x)$	$r(\theta)$
least-squares	$(\theta^T x - y)^2$	$\theta^T x$	0
ridge regression	$(\theta^T x - y)^2$	$\theta^T x$	$\lambda \|\theta\|_2^2$
lasso	$(\theta^T x - y)^2$	$\theta^T x$	$\lambda \|\theta\|_1$
logistic classifier	$\log(1 + \exp(-y\theta^T x))$	$\mathbf{sign}(\theta^T x)$	0
SVM	$(1 - y\theta^T x)_+$	$\mathbf{sign}(\theta^T x)$	$\lambda \|\theta\|_2^2$

- $\lambda > 0$ scales regularization
- all lead to convex fitting problems

Example

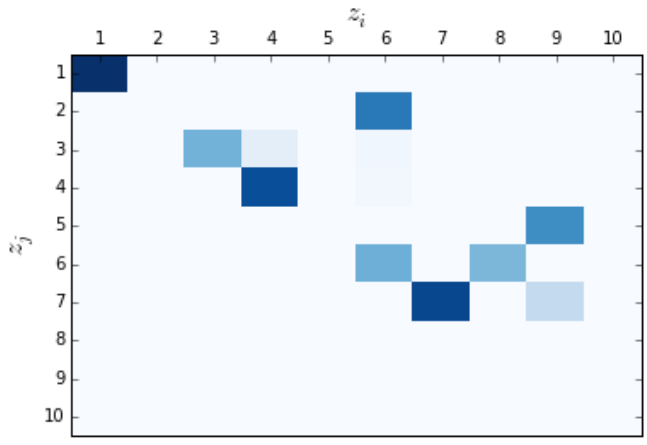
- original (boolean) features $z \in \{0,1\}^{10}$
- (boolean) outcome $y \in \{-1,1\}$
- new feature vector $x \in \{0,1\}^{55}$ contains all products $z_i z_j$ (co-occurence of pairs of original features)
- use logistic loss, ℓ_1 regularizer
- training data has $m = 200$ examples; test on 100 examples

Example


Train error
Test error
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
10^{-4}
10^{-3}
10^{-2}
10^{-1}
10^{0}
λ

Example

selected features $z_i z_j$, $\lambda = 0.01$



Outline

Homework Review

Penalty function approximation

Regression

Model Fitting

Maximum likelihood estimation

Maximum likelihood estimation

- **parametric distribution estimation:** choose from a family of densities $p_x(y)$, indexed by a parameter x (often denoted θ)
- we take $p_x(y) = 0$ for invalid values of x
- $p_x(y)$, as a function of x, is called **likelihood function**
- $l(x) = \log p_x(y)$, as a function of x, is called **log-likelihood function**

- **maximum likelihood estimation (MLE):** choose x to maximize $p_x(y)$ (or $l(x)$)
- a convex optimization problem if $\log p_x(y)$ is concave in x for fixed y
- not the same as $\log p_x(y)$ concave in y for fixed x, *i.e.*, $p_x(y)$ is a family of log-concave densities

Linear measurements with IID noise

linear measurement model

$$y_i = a_i^T x + v_i, \quad i = 1, \ldots, m$$

- $x \in \mathbf{R}^n$ is vector of unknown parameters
- v_i is IID measurement noise, with density $p(z)$
- y_i is measurement: $y \in \mathbf{R}^m$ has density $p_x(y) = \prod_{i=1}^m p(y_i - a_i^T x)$

maximum likelihood estimate: any solution x of

$$\text{maximize} \quad l(x) = \sum_{i=1}^m \log p(y_i - a_i^T x)$$

(y is observed value)

Examples

- Gaussian noise $\mathcal{N}(0, \sigma^2)$: $p(z) = (2\pi\sigma^2)^{-1/2} e^{-z^2/(2\sigma^2)}$,

$$l(x) = -\frac{m}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{m}(a_i^T x - y_i)^2$$

  ML estimate is least-squares solution

- Laplacian noise: $p(z) = (1/(2a))e^{-|z|/a}$,

$$l(x) = -m\log(2a) - \frac{1}{a}\sum_{i=1}^{m}|a_i^T x - y_i|$$

  ML estimate is ℓ_1-norm solution

- uniform noise on $[-a, a]$:

$$l(x) = \begin{cases} -m\log(2a) & |a_i^T x - y_i| \le a, \quad i = 1, \ldots, m \\ -\infty & \text{otherwise} \end{cases}$$

  ML estimate is any x with $|a_i^T x - y_i| \le a$

Logistic regression

- random variable $y \in \{0, 1\}$ with distribution

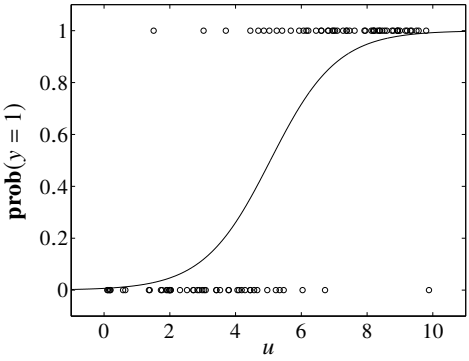
$$p = \mathbf{prob}(y = 1) = \frac{\exp(a^T u + b)}{1 + \exp(a^T u + b)}$$

- a, b are parameters; $u \in \mathbf{R}^n$ are (observable) explanatory variables
- estimation problem: estimate a, b from m observations (u_i, y_i)
- log-likelihood function (for $y_1 = \cdots = y_k = 1$, $y_{k+1} = \cdots = y_m = 0$):

$$\begin{aligned} l(a, b) &= \log\left(\prod_{i=1}^{k} \frac{\exp(a^T u_i + b)}{1 + \exp(a^T u_i + b)} \prod_{i=k+1}^{m} \frac{1}{1 + \exp(a^T u_i + b)}\right) \\ &= \sum_{i=1}^{k} (a^T u_i + b) - \sum_{i=1}^{m} \log(1 + \exp(a^T u_i + b)) \end{aligned}$$

concave in a, b

Example



- $n = 1$, $m = 50$ measurements; circles show points (u_i, y_i)
- solid curve is ML estimate of $p = \exp(au + b)/(1 + \exp(au + b))$

Gaussian covariance estimation

- fit Gaussian distribution $\mathcal{N}(0, \Sigma)$ to observed data $y_1, \ldots, y_N$
- log-likelihood is

$$\begin{aligned} l(\Sigma) &= \frac{1}{2} \sum_{k=1}^{N} \left(-2\pi n - \log\det\Sigma - y^T \Sigma^{-1} y \right) \\ &= \frac{N}{2} \left(-2\pi n - \log\det\Sigma - \mathbf{tr}\, \Sigma^{-1} Y \right) \end{aligned}$$

  with $Y = (1/N) \sum_{k=1}^{N} y_k y_k^T$, the empirical covariance

- l is **not** concave in Σ (the $\log\det\Sigma$ term has the wrong sign)
- with no constraints or regularization, MLE is empirical covariance $\Sigma^{\mathsf{ml}} = Y$

Change of variables

- change variables to $S = \Sigma^{-1}$
- recover original parameter via $\Sigma = S^{-1}$
- S is the **natural parameter** in an **exponential family** description of a Gaussian
- in terms of S, log-likelihood is

$$l(S) = \frac{N}{2}\left(-2\pi n + \log\det S - \mathbf{tr}\, SY\right)$$

  which is **concave**

- (a similar trick can be used to handle nonzero mean)