Logistic Regression

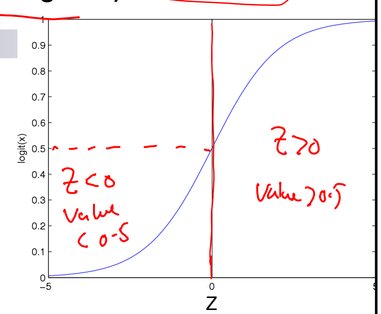
Machine Learning – 10701/15781

Carlos Guestrin
Carnegie Mellon University

September 28th, 2009

Logistic Regression

Logistic function (or Sigmoid): $\frac{1}{1+exp(-z)}$



- Learn P(Y|**X**) directly!
  - Assume a particular functional form
  - Sigmoid applied to a linear function of the data:

Logistic regression

$$P(Y=1|X) = \frac{1}{1+\exp(w_0 + \sum_{i=1}^{n} w_i X_i)}$$

regression → ℝ

Logistic regression ∈ [0,1]

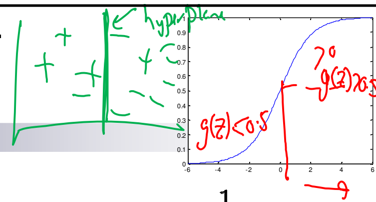
Params: $w_0, w_1, \ldots, w_n$

like applying logistic regression to linear regression function

x_i

Features can be discrete or continuous!

Logistic Regression – a Linear classifier



$$g(w_0 + \sum_i w_i x_i) = \frac{1}{1 + e^{w_0 + \sum_i w_i x_i}}$$

$X = \{X_1, \ldots, X_n\}$ each X_i can be anything

$P(Y=1 \mid X, w) < 0.5$ binary class

$w_0 + \sum_i w_i x_i > 0$

predict $Y=0$

$w_0 + \sum_i w_i x_i = 0$ ← hyperplane

$w_0 + \sum_i w_i x_i < 0$

$P(Y=1 \mid X, w) > 0.5$

predict $Y=1$

$g(z) < 0.5$ > 0 $g(z) \geq 0.5$

Logistic regression for more than 2 classes

- Logistic regression in more general case, where $Y \in \{y_1,\ldots,y_R\}$

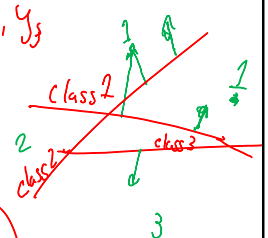
3 classes: y_1, y_2, y_3

$P(Y=1 \mid X) \propto e^{w_{10} + \sum_i w_{1i} X_i}$

$P(Y=2 \mid X) \propto e^{w_{20} + \sum_i w_{2i} X_i}$

$\vdots$

$P(Y=3 \mid X) = 1 - P(Y=1 \mid X) - P(Y=2 \mid X)$



Logistic regression more generally

- Logistic regression in more general case, where $Y \in \{y_1,\ldots,y_R\}$

learn w_{ki}

for $k<R$

$$P(Y=y_k|X) = \frac{\exp(w_{k0}+\sum_{i=1}^n w_{ki}X_i)}{1+\sum_{j=1}^{R-1}\exp(w_{j0}+\sum_{i=1}^n w_{ji}X_i)}$$

for $k=R$ (normalization, so no weights for this class)

$$P(Y=y_R|X) = \frac{1}{1+\sum_{j=1}^{R-1}\exp(w_{j0}+\sum_{i=1}^n w_{ji}X_i)}$$

Features can be discrete or continuous!

Loss functions: Likelihood v. Conditional Likelihood

$P(X,Y) = P(X)P(Y|X)$ chain rule $= P(Y)P(X|Y)$

- Generative (e.g. Naïve Bayes) Loss function:
  Data likelihood

vector of features

class

$$\ln P(\mathcal{D} \mid \mathbf{w}) = \sum_{j=1}^N \ln P(\mathbf{x}^j, y^j \mid \mathbf{w})$$
$$= \underbrace{\sum_{j=1}^N \ln P(y^j \mid \mathbf{x}^j, \mathbf{w})}_{\text{conditional likelihood}} + \underbrace{\sum_{j=1}^N \ln P(\mathbf{x}^j \mid \mathbf{w})}_{\text{if you have a generative model}}$$

- Discriminative models cannot compute $P(\mathbf{x}^j|\mathbf{w})$!
- But, discriminative (e.g. logistic regression) loss function:
  Conditional Data Likelihood

only this part

$$\ln P(\mathcal{D}_Y \mid \mathcal{D}_\mathbf{X}, \mathbf{w}) = \sum_{j=1}^N \ln P(y^j \mid \mathbf{x}^j, \mathbf{w})$$

  - Doesn't waste effort learning P(X) – focuses on P(Y|**X**) all that matters for classification

$\ln \frac{a}{b} = \ln a - \ln b$ $\qquad \ln e^{TA's} = TA's$ $\qquad \ln 1 = 0$

Expressing Conditional Log Likelihood

Only going to derive for binary class.
multiclass is very similar

$$l(\mathbf{w}) \equiv \underbrace{\sum_j \ln P(y^j|\mathbf{x}^j, \mathbf{w})}$$

$y^j \in \{0,1\}$

$$P(Y=0|\mathbf{X},\mathbf{w}) = \frac{1}{1+exp(w_0+\sum_i w_i X_i)}$$

$$P(Y=1|\mathbf{X},\mathbf{w}) = \frac{exp(w_0+\sum_i w_i X_i)}{1+exp(w_0+\sum_i w_i X_i)}$$

$$= \sum_j \begin{matrix} (\text{if } y^j=0) \rightarrow \ln P(Y=0|x^j,w) \\ (\text{if } y^j=1) \rightarrow \ln P(Y=1|x^j,w) \end{matrix}$$

$$l(\mathbf{w}) = \sum_j \underbrace{y^j \ln P(\cancel{y^j}\,Y = 1|\mathbf{x}^j, \mathbf{w})}_{\text{contributes when } y_j=1} + \underbrace{(1-y^j) \ln P(\cancel{y^j}\,Y = 0|\mathbf{x}^j, \mathbf{w})}_{\text{when } y_j=0}$$

$$= \sum_j y_j \left[\ln \frac{e^{w_0+\sum_i w_i x_i}}{1+e^{w_0+\sum_i w_i x_i}}\right] + (1-y_j)\left(\frac{1}{1+e^{w_0+\sum_i w_i x_i}}\right)$$

$$= \sum_j y_j(w_0+\sum_i w_i x_i) \underbrace{- y_j \ln(1+e^{w_0+\sum_i w_i x_i}) - (1-y_j)\ln(1+e^{w_0+\sum_i w_i x_i})}_{-\ln(1+e^{w_0+\sum_i w_i x_i})}$$

Maximizing Conditional Log Likelihood

$$P(Y=0|X,W) = \frac{1}{1+exp(w_0+\sum_i w_i X_i)}$$

$$P(Y=1|X,W) = \frac{exp(w_0+\sum_i w_i X_i)}{1+exp(w_0+\sum_i w_i X_i)}$$

maximize

$$l(\mathbf{w}) \equiv \ln \prod_j P(y^j|\mathbf{x}^j, \mathbf{w})$$

$$= \sum_{j=1}^{N} y^j(w_0 + \sum_i^n w_i x_i^j) - \ln(1 + exp(w_0 + \sum_i^n w_i x_i^j))$$

Multiclass → still concave, but eqn is slightly longer...

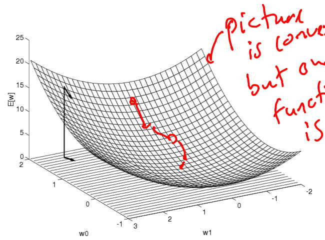
Good news: $l(\mathbf{w})$ is concave function of $\mathbf{w}$ ⇒ no locally optimal solutions

Bad news: no closed-form solution to maximize $l(\mathbf{w})$

Good news: concave functions "easy" to optimize

Optimizing concave function – Gradient ascent

- Conditional likelihood for Logistic Regression is concave $\Rightarrow$ Find optimum with gradient ascent



picture is convex but our function is concave

Gradient: $\nabla_{\mathbf{w}} l(\mathbf{w}) = [\frac{\partial l(\mathbf{w})}{\partial w_0}, \ldots, \frac{\partial l(\mathbf{w})}{\partial w_n}]'$

Learning rate, η>0

step size (learning rate)

Update rule: $\Delta \mathbf{w} = \eta \nabla_{\mathbf{w}} l(\mathbf{w})$

$$w_i^{(t+1)} \leftarrow w_i^{(t)} + \eta \frac{\partial l(\mathbf{w})}{\partial w_i}$$

- Gradient ascent is simplest of optimization approaches
  - e.g., Conjugate gradient ascent much better (see reading)

Maximize Conditional Log Likelihood: Gradient ascent

$\frac{\partial}{\partial x} \ln f(x) = \frac{f'(x)}{f(x)}$ $\qquad$ $\frac{\partial}{\partial x} e^{ax} = a e^{ax}$

$$l(\mathbf{w}) = \sum_j y^j (w_0 + \sum_i^n w_i x_i^j) - \ln(1 + exp(w_0 + \sum_i^n w_i x_i^j))$$

$$\frac{\partial l(w)}{\partial w_i} = \sum_j \underbrace{\frac{\partial}{\partial w_i} y^j (w_0 + \sum_i w_i x_i^j)} - \underbrace{\frac{\partial}{\partial w_i} \ln (1 + e^{w_0 + \sum w_i x_i^j})}$$

$$= \sum_j y^j x_i^j - \frac{x_i^j e^{w_0 + \sum w_i x_i^j}}{1 + e^{w_0 + \sum w_i x_i^j}}$$

$$= \sum_j x_i^j \Big(y^j - \underbrace{\frac{e^{w_0 + \sum_i w_i x_i^j}}{1 + e^{w_0 + \sum w_i x_i^j}}}_{P(Y=1 \mid x^j, w)} \Big)$$

$$\frac{\partial l(w)}{\partial w_i} = \sum_j x_i^j (y^j - P(Y=1 \mid x^j, w))$$

Gradient Descent for LR

most correctly $\ell(w)$

Gradient ascent algorithm: iterate until "change" < ε
you can believe in

$$w_0^{(t+1)} \leftarrow w_0^{(t)} + \eta \sum_j [y^j - \hat{P}(Y^j = 1 \mid \mathbf{x}^j, \mathbf{w}^{(t)})]$$

For i=1,…,n,

$$w_i^{(t+1)} \leftarrow w_i^{(t)} + \eta \sum_j x_i^j [y^j - \underbrace{\hat{P}(Y^j = 1 \mid \mathbf{x}^j, \mathbf{w}^{(t)})}_{\text{predict what } w^{(t)} \text{ thinks data point } j \text{ should be}}]$$

repeat

That's all M(C)LE. How about MAP?

$$p(\mathbf{w} \mid Y, \mathbf{X}) \propto \underbrace{P(Y \mid \mathbf{X}, \mathbf{w})}_{\text{likelihood}} \underbrace{p(\mathbf{w})}_{\text{prior over params}}$$

- One common approach is to define priors on **w**
  - Normal distribution, zero mean, identity covariance
  - "Pushes" parameters towards zero
- Corresponds to ***Regularization***
  - Helps avoid very large weights and overfitting
  - More on this later in the semester
- MAP estimate

$$\mathbf{w}^* = \arg\max_{\mathbf{w}} \ln \left[p(\mathbf{w}) \prod_{j=1}^{N} P(y^j \mid \mathbf{x}^j, \mathbf{w}) \right]$$

ln ab = ln a + ln b

M(C)AP as Regularization

$$\ln \left[p(\mathbf{w}) \prod_{j=1}^{N} P(y^j \mid \mathbf{x}^j, \mathbf{w}) \right]$$

$$\underbrace{p(\mathbf{w}) = \prod_i \frac{1}{\kappa\sqrt{2\pi}} \; e^{\frac{-w_i^2}{2\kappa^2}}}_{\text{zero mean gaussian prior}}$$

variance κ^2

$$= \underbrace{\ln p(w)} + \underbrace{\sum_j \ln P(y^j \mid x^j, w)}_{\text{tired of this you know what it is}}$$

$$= \ln \prod_i \frac{1}{\kappa\sqrt{2\pi}} e^{\frac{-w_i^2}{2\kappa^2}}$$

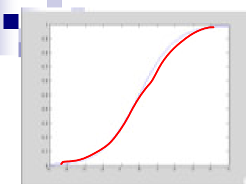
$$= \sum_i \ln \frac{1}{\kappa\sqrt{2\pi}} + \sum_i \frac{-w_i^2}{2\kappa^2}$$

const. doesn't depend on w → ignore

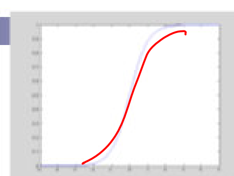
$$\Rightarrow \underbrace{\sum_j \ln P(y^j \mid x^j, w)}_{\text{as before}} \; \underbrace{- \sum_i \frac{w_i^2}{2\kappa^2}}_{\text{penalty for large } w_i}$$

Penalizes high weights, also applicable in linear regression

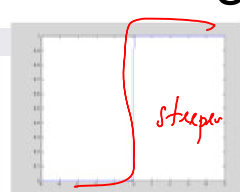
Large parameters → Overfitting



$\frac{1}{1+e^{-x}}$ $w_1 = -1$



$\frac{1}{1+e^{-2x}}$ $w_1 = -2$



$\frac{1}{1+e^{-100x}}$ $w_1 = -100$

logistic regression without regularization will give huge weights → overfitting

- If data is linearly separable, weights go to infinity
- Leads to overfitting: ↳ if ∃ hyperplane that separates 1 & 0

↑ linearly separable

intuitively may not want this hyperplane

- Penalizing high weights can prevent overfitting…
  - again, more on this later in the semester

Gradient of M(C)AP

$\ln ab = \ln a + \ln b$

$\frac{\partial}{w_i} w_i^2 = 2w_i$

$$\frac{\partial}{\partial w_i} \ln \left[p(\mathbf{w}) \prod_{j=1}^{N} P(y^j \mid \mathbf{x}^j, \mathbf{w}) \right] \qquad p(\mathbf{w}) = \prod_i \frac{1}{\kappa\sqrt{2\pi}} \; e^{\frac{-w_i^2}{2\kappa^2}}$$

$$= \underbrace{\frac{\partial}{\partial w_i} \ln P(w)} + \underbrace{\frac{\partial}{\partial w_i} \ln \prod_j P(y^j \mid x^j, w)}_{\text{as before}}$$

$$\frac{\partial}{\partial w_i}\left[\text{const.} + \sum_i \frac{-w_i^2}{2\kappa^2}\right] = \frac{-w_i}{\kappa^2}$$

extra term pushes w_i towards $\emptyset$

if $w_i \gg 0 \Rightarrow \frac{-w_i}{\kappa^2}$ is very negative
$w_i \ll 0 \Rightarrow \frac{-w_i}{\kappa^2}$ is very positive
} push w_i to $\emptyset$

MLE vs MAP

in practice, ALWAYS REGULARIZE YOUR LR

Typically don't regularize w_0

- Maximum conditional likelihood estimate

$$\mathbf{w}^* = \arg\max_{\mathbf{w}} \ln \left[\prod_{j=1}^{N} P(y^j \mid \mathbf{x}^j, \mathbf{w}) \right]$$

$$\boxed{w_i^{(t+1)} \leftarrow w_i^{(t)} + \eta \sum_j x_i^j [y^j - \hat{P}(Y^j = 1 \mid \mathbf{x}^j, \mathbf{w}^{(t)})]}$$

update rule gradient ascent (regularized)

- Maximum conditional a posteriori estimate

$$\mathbf{w}^* = \arg\max_{\mathbf{w}} \ln \left[p(\mathbf{w}) \prod_{j=1}^{N} P(y^j \mid \mathbf{x}^j, \mathbf{w}) \right]$$

extra term $\lambda = \frac{1}{\kappa^2}$

$$\boxed{w_i^{(t+1)} \leftarrow w_i^{(t)} + \eta \left\{ -\lambda w_i^{(t)} + \sum_j x_i^j [y^j - \hat{P}(Y^j = 1 \mid \mathbf{x}^j, \mathbf{w}^{(t)})] \right\}}$$

Logistic regression v. Naïve Bayes

- Consider learning f: X → Y, where
  - X is a vector of real-valued features, < X1 … Xn >
  - Y is boolean
- Could use a Gaussian Naïve Bayes classifier
  - assume all X_i are conditionally independent given Y
  - model $P(X_i \mid Y = y_k)$ as Gaussian $N(\mu_{ik},\sigma_i)$
  - model P(Y) as Bernoulli(θ,1-θ)

for this special case GNB can be represented by a LR

mean

assume the variance indep. of class

- What does that imply about the form of P(Y|X)?

$$P(Y=1|X=<X_1,...X_n>) = \frac{1}{1+exp(w_0+\sum_i w_i X_i)}$$

Cool!!!!

Derive form for P(Y|X) for continuous X_i

$= P(Y)\cancel{\frac{P(X|Y)}{P(X)}}$

$$P(Y=1|X) = \frac{P(Y=1)P(X|Y=1)}{P(Y=1)P(X|Y=1)+P(Y=0)P(X|Y=0)}$$

$$= \frac{1}{1+\frac{P(Y=0)P(X|Y=0)}{P(Y=1)P(X|Y=1)}}$$

$$= \frac{1}{1+\exp(\ln\frac{P(Y=0)P(X|Y=0)}{P(Y=1)P(X|Y=1)})}$$

only assumption thus far is NB

$$= \frac{1}{1+\exp(\ \underbrace{(\ln\frac{1-\theta}{\theta})}_{\text{reminds me of } w_0}+\boxed{\underbrace{\sum_i \ln\frac{P(X_i|Y=0)}{P(X_i|Y=1)}}_{\text{reminds me of } w_i X_i}})}$$

let's check

Ratio of class-conditional probabilities

$\ln \frac{a}{b} = \ln a - \ln b \qquad \ln e^{\text{something}} = \text{something}$

formula for GNB,

$$\boxed{P(X_i = x \mid Y = y_k) = \frac{1}{\sigma_i \sqrt{2\pi}} \; e^{\frac{-(x-\mu_{ik})^2}{2\sigma_i^2}}}$$

σ_i ← class indep

$$\ln \frac{P(X_i|Y=0)}{P(X_i|Y=1)} \quad (y_k)$$

$$\ln \frac{\cancel{\frac{1}{\sigma_i\sqrt{2\pi}}}\; e^{-\frac{(x_i-\mu_{i0})^2}{2\sigma_i^2}}}{\cancel{\frac{1}{\sigma_i\sqrt{2\pi}}}\; e^{-\frac{(x_i-\mu_{i1})^2}{2\sigma_i^2}}}$$

$$= -\frac{(x_i-\mu_{i0})^2}{2\sigma_i^2} + \frac{(x_i-\mu_{i1})^2}{2\sigma_i^2}$$

$$= \frac{-\cancel{x_i^2} + 2\mu_{i0}x_i - \mu_{i0}^2 + \cancel{x_i^2} - 2\mu_{i1}x_i + \mu_{i1}^2}{2\sigma_i^2}$$

$$= \left(\underbrace{\frac{2\mu_{i0}-2\mu_{i1}}{2\sigma_i}}_{w_i}\right) x_i + \underbrace{\frac{\mu_{i1}^2-\mu_{i0}^2}{2\sigma_i^2}}_{\text{part of } w_0}$$

goal.
$\ln \frac{P(X_i|Y=0)}{P(X_i|Y=1)}$
to look like
$w_i X_i$

Derive form for P(Y|X) for continuous X_i

$$P(Y=1|X) = \frac{P(Y=1)P(X|Y=1)}{P(Y=1)P(X|Y=1) + P(Y=0)P(X|Y=0)}$$

$$= \frac{1}{1 + \exp\left(\left(\ln \frac{1-\theta}{\theta}\right) + \boxed{\sum_i \ln \frac{P(X_i|Y=0)}{P(X_i|Y=1)}}\right)}$$

$$\boxed{\sum_i \left(\frac{\mu_{i0}-\mu_{i1}}{\sigma_i^2} X_i + \frac{\mu_{i1}^2 - \mu_{i0}^2}{2\sigma_i^2} \right)}$$

$$P(Y=1|X) = \frac{1}{1+\exp(w_0 + \sum_{i=1}^n w_i X_i)}$$

$$w_0 = \ln\left(\frac{1-\theta}{\theta}\right) + \sum_i \frac{\mu_{i1}^2 - \mu_{i0}^2}{2\sigma_i^2}$$

Gaussian Naïve Bayes v. Logistic Regression

Set of Gaussian Naïve Bayes parameters (feature variance independent of class label)

transform: ∃ ws for LR that are equivalent

Set of Logistic Regression parameters

fewer params in LR than GNB
∤ can find $\mu_{i0}, \mu_{i1}, \theta$ to make LR's w into GNB

- Representation equivalence
  - **But only in a special case!!!** (GNB with class-independent variances)
- But what's the difference???
- **LR makes no assumptions about** P(**X**|Y) **in learning**!!!
- **Loss function!!!**
  - Optimize different functions ⇒ Obtain different solutions

LR : $P(Y|X,w)$
NB : $P(Y,X|w)$

Naïve Bayes vs Logistic Regression

Consider Y boolean, X_i continuous, X=<X_1 ... X_n>

Number of parameters:
- NB: 4n +1
- LR: n+1

Estimation method:
- NB parameter estimates are uncoupled
- LR parameter estimates are coupled

G. Naïve Bayes vs. Logistic Regression 1

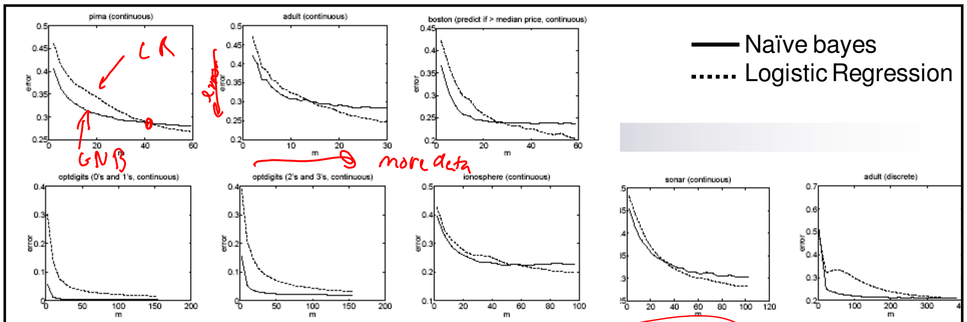
[Ng & Jordan, 2002]

- Generative and Discriminative classifiers
- Asymptotic comparison (# training examples → infinity)
  - when model correct
    GNB independences
    - GNB (with class independent variances) and LR produce identical classifiers
  - when model incorrect
    GNB
    - LR is less biased – does not assume conditional independence
      - **therefore LR expected to outperform GNB**

G. Naïve Bayes vs. Logistic Regression 2

[Ng & Jordan, 2002]

- Generative and Discriminative classifiers
- Non-asymptotic analysis
  - convergence rate of parameter estimates, n = # of attributes in X
    - Size of training data to get close to infinite data solution
    - GNB needs *O*(log n) samples   n features (because of indep.)
    - LR needs *O*(n) samples   need more data
  - **GNB converges more quickly to its (perhaps less helpful) asymptotic estimates**



not enough data for LR to beat GNB

Some experiments from UCI data sets

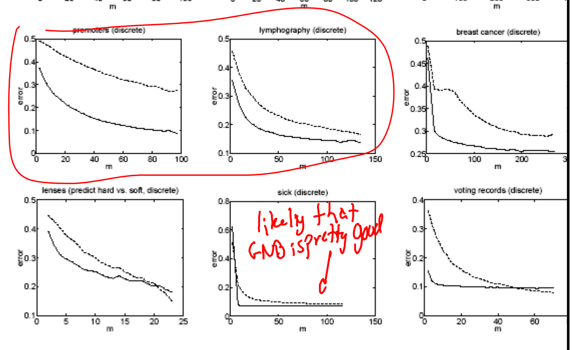


Figure 1: Results of 15 experiments on datasets from the UCI Machine Learnin repository. Plots are of generalization error vs. m (averaged over 1000 rando train/test splits). Dashed line is logistic regression; solid line is naive Bayes.

What you should know about Logistic Regression (LR)

- Gaussian Naïve Bayes with class-independent variances representationally equivalent to LR
  - Solution differs because of objective (loss) function
- In general, NB and LR make different assumptions
  - NB: Features independent given class ! assumption on P(**X**|Y)
  - LR: Functional form of P(Y|**X**), no assumption on P(**X**|Y)
- LR is a linear classifier
  - decision rule is a hyperplane
- LR optimized by conditional likelihood
  - no closed-form solution
  - concave ! global optimum with gradient ascent
  - Maximum conditional a posteriori corresponds to regularization
- Convergence rates
  - GNB (usually) needs less data
  - LR (usually) gets to better solutions in the limit