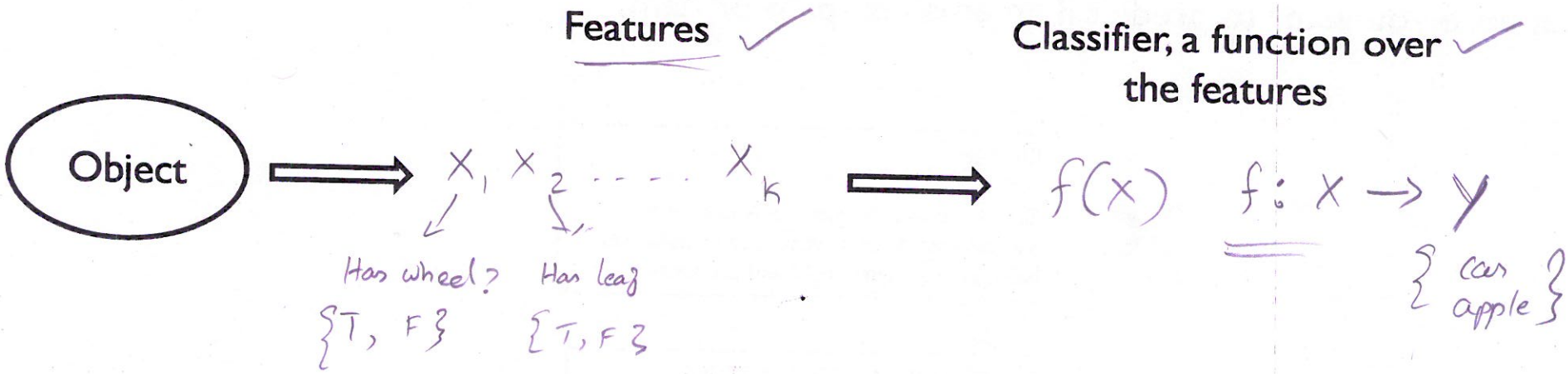
Learning to Predict



Two main questions in ML:

1) How do we get features?

- Domain experts / Problem study to give features
- Deep learning allows learning features too! from raw input

2) How do we obtain the classifier function?

linear function: $\sum w_i x_i$,

→ linear fun

non-linear functions

General Naïve Bayes Model

> Suppose you want to predict if an email is spam or ham.

> What are your outputs?

$Y = \{ \text{Spam}, \text{Ham} \}$

Dear Sir.

First, I must solicit your confidence in this transaction, this is by virture of its nature as being utterly confidencial and top secret. …

What are your features?

1. has-nigerian-prince — $F_1 = \{T, F\}$
2. has-$$$ — $F_2 = \{T, F\}$
3. is-grammatical — F_3
4. ALL-CAPS? — F_4

TO BE REMOVED FROM FUTURE MAILINGS, SIMPLY REPLY TO THIS MESSAGE AND PUT "REMOVE" IN THE SUBJECT.

99 MILLION EMAIL ADDRESSES FOR ONLY $99

Ok, Iknow this is blatantly OT but I'm beginning to go insane. Had an old Dell Dimension XPS sitting in the corner and decided to put it to use, I know it was working pre being stuck in the corner, but when I plugged it in, hit the power nothing happened.

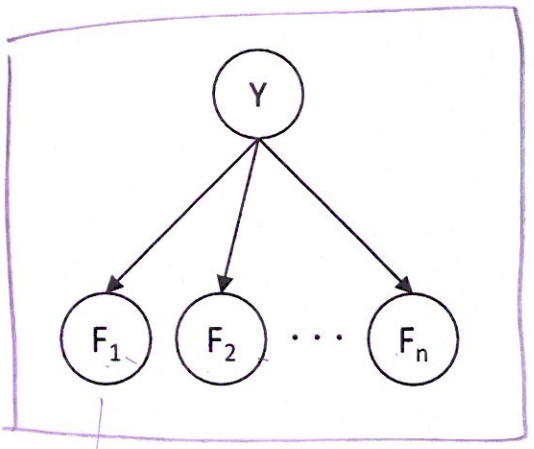
Let's denote these by random variables F_i.

Naïve Bayes is a Bayesian Network!

- Assume all features are independent effects of label.
  - We only have to specify how each feature depends on the class

$$P(Y, F_1 \ldots F_n) = Pr(Y) \times Pr(F_1|Y) \times Pr(F_2|Y) \times \ldots \times Pr(F_n|Y)$$

$$= Pr(Y) \prod_{i=1}^{n} Pr(F_i|Y)$$



- What are the CPTs associated with this network?

Y	Pr(Y)
Spam	
Ham	

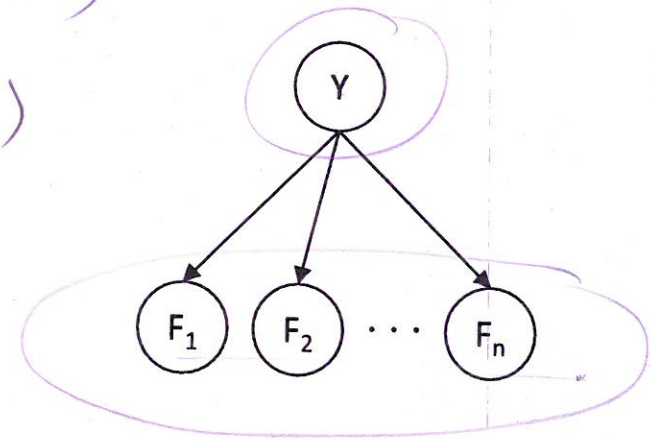
Y	F_1	Pr(F_1|Y)
Spam	T	
Spam	F	
Ham	T	
Ham	F	

Naïve Bayes is a Bayesian Network!

- What probability do we want for classification?

$$Pr(Y = Spam \mid F_1 = f_1, F_2 = f_2 \ldots F_n = f_n)$$

$$Pr(Y = Ham \mid F_1 = f_1 \ldots \quad)$$



- How to get the conditional from the joint for this network? Inference!

$$Pr(Y \mid F_1 \ldots F_n) = \frac{Pr(Y, F_1 \ldots F_n)}{Pr(F_1 \ldots F_n)}$$

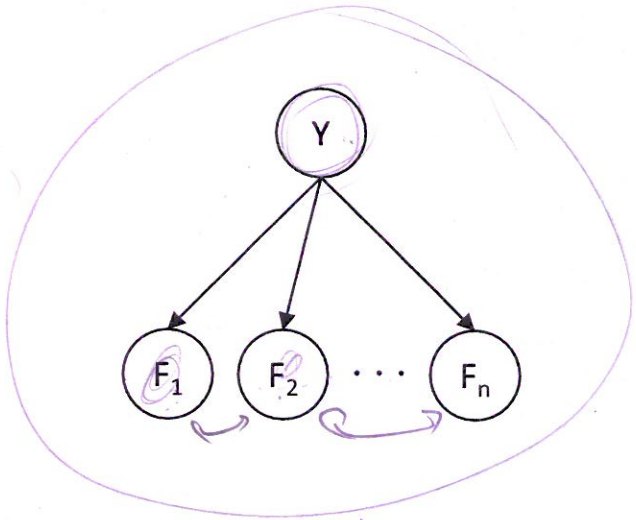
$$\underset{Y=\{S,H\}}{\operatorname{argmax}} Pr(Y \mid F_1 \ldots F_n) = \underset{Y}{\operatorname{argmax}} \frac{Pr(Y, F_1 \ldots F_n)}{Pr(F_1 \ldots F_n)} = \underset{Y}{\operatorname{argmax}} Pr(Y) \prod_{i=1}^{n} Pr(F_i \mid Y)$$

Naïve Bayes is a Bayesian Network!

- How do we get the conditional probabilities? Estimate from training data.

~~$Pr(y) = \#(y)$~~

~~$Pr(F_i|y) = ?$~~



$$Pr(Y = \text{spam}) = \frac{\#(Y = \text{spam})}{N}$$ ← total # examples

$$Pr(F_i = T \mid Y = \text{spam}) = \frac{\#(F_i = T,\ Y = \text{spam})}{\#(Y = \text{spam})}$$

An example: Bag-of-words Naïve Bayes

- Data:
  - Collection of emails, labeled spam or ham
  - Note: someone has to hand label all this data!
  - Split into training, held-out, test sets
- Features
  - Words that occur in the document.
- Classifier
  - Bayesian Network
  - Label conditioned on feature variables (spam vs. ham)
  - Assume features are conditionally independent given class label

Dear Sir.

First, I must solicit your confidence in this transaction, this is by virture of its nature as being utterly confidencial and top secret. …

TO BE REMOVED FROM FUTURE MAILINGS, SIMPLY REPLY TO THIS MESSAGE AND PUT "REMOVE" IN THE SUBJECT.

99 MILLION EMAIL ADDRESSES FOR ONLY $99

Ok, Iknow this is blatantly OT but I'm beginning to go insane. Had an old Dell Dimension XPS sitting in the corner and decided to put it to use, I know it was working pre being stuck in the corner, but when I plugged it in, hit the power nothing happened.

An example: Bag-of-words Naïve Bayes

- BN Variables? $Y = \{S, H\}$

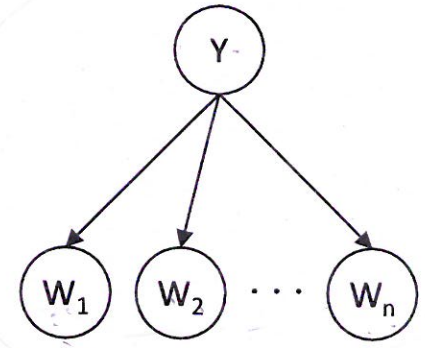
W_i = word in position i

e.g W_1 = Nigerian

W_2 = Prince

W_3 = wants

$W_i = \{a, ian, the, \ldots, Zebra\}$



Email.

Nigerian prince wants to send you $1M.

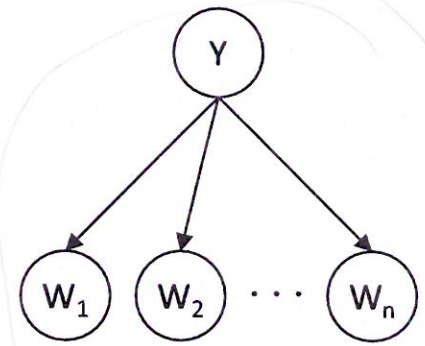
- BN CPTs?

- Ignore for now!

An example: Bag-of-words Naïve Bayes

- Joint Distribution?

$$Pr(Y, W_1, \ldots W_n) = Pr(Y) \prod_{i=1}^{n} Pr(W_i | Y)$$



- What do we want for prediction?

$$Pr(Y | W_1 \ldots W_n)$$

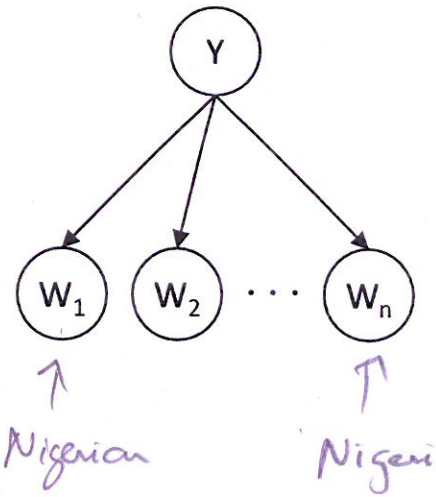
- Use Bayes Rule

$$\underset{y}{\text{argmax}} \; Pr(Y) \prod_{i=1}^{n} Pr(W_i | Y)$$

An example: Bag-of-words Naïve Bayes

- Use training data to estimate!

$P(Y)$ = $\frac{\#(Y = \text{spam})}{N}$, $\frac{\#(Y = H)}{N}$ → # emails.



$P(W|\text{spam})$

$\Pr(W_i|\text{Spam}) = \frac{\#(W = w \text{ , spam})}{\#\text{ Spam emails.}}$ → # of spam emails in which w was found

$\Pr(W_1 = \text{Nigeria}|\text{spam}) = \Pr(W_n = \text{Nig}|\text{Spam})$

$P(W|\text{ham})$

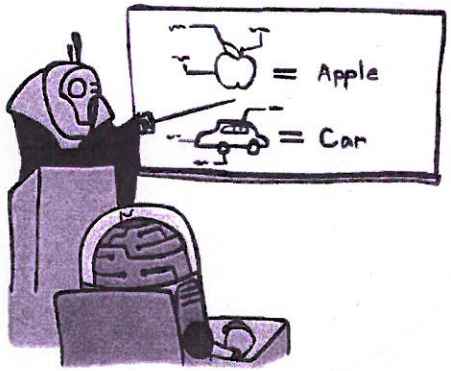
- Maximum likelihood estimation
- Max *a posteriori* estimation → expectations on how the probabilities should look like.

MAP

How do we know our classifier is any good?

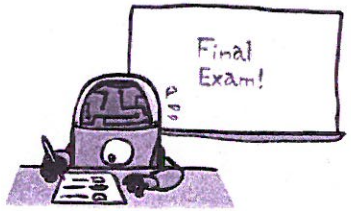
Experimental Cycle

Train



Held-Out Data

Test Data



Evaluation Metrics:

Accuracy = $\frac{\#(\text{correct labels})}{\#(\text{labels})}$

multiple classes

Precision = $\frac{\#(\text{model} = \text{spam}, \text{truth} = \text{spam})}{\#(\text{model} = \text{spam})}$

spam

Recall = $\frac{\#(\text{model}=S, \text{truth}=S)}{\#(\text{truth}=S)}$

F-measure = $\frac{2 \cdot PR}{P+R}$

Harmonic mean of P & R

Issues: Independence assumptions

- Need more features– words aren't enough!
  - Have other people just gotten the same email?
  - Is the email in ALL CAPS?
  - Do inline URLs point where they say they point?

- Can add these information sources as new variables in the NB model
  - But will be violating independence assumptions even more.
  - Example:
    - "New" and "York" are dependent even when the class is observed.
    - Knowing "York" tells me New is likely to be present in the email.
    - If you P("New"|C) and P("York"|C) both contribute independent evidence.

- There are classifiers which let you easily add arbitrary features more easily.
  - We'll see one such next.

Discriminative Classification

- Bayesian networks are generative models
  - They model the joint distribution of the instances and the labels.
  - Generative in the sense that they can gen. samples.
- Why model joint P(Y, X)?
  - After all we only care about P(Y|X)

- We are interested in selecting the best Y given the evidence X i.e, P(Y|X)

  - Models that directly estimate this are called discriminative classifiers!

e.g Logistic Regression

Discriminative Classification = Conditional Probability Estimation

- Let's learn a function that can predict P(Y|X) directly!

- Assume that P(Y|X) is a parameterized function of some form.

$P(Y|x) \approx h_W(x, y)$ → $h_W(x) = 1$ if x is spam, 0 if x is h

For spam email (x^i, y^i) $h_W(x^i, y^i) = P_r(y = \text{spam} \mid x^i) = 1$

- Find the "best" values for these parameters using the training data.

$h_W(x)$ → has free parameters

$W = (w_1, w_2 \ldots, w_n)$

$F_1 \quad F_2 \ldots F_n$

- Logistic regression is one example.

Logistic Regression Recipe: For binary classification

- Assume that there is a function that approximates P(Y|X)

$$h_w(x)$$

- We will jump a couple of steps to arrive at this form.

- First, what constraints should this function satisfy?

$$h_w(x) \approx P_\theta(Y|x)$$

$$\Rightarrow \quad 0 \leq h_w(x) \leq 1$$

Training: Setting an objective function

- Select a set of parameters that maximize the probability of the training data.

$$\text{Train} = \{(y_c^1, x^1), (y_c^2, x^2), \ldots, (y_c^\ell, x^\ell)\}$$

Email 1 Email 2 ...

- Probability of the training data

$$\prod_{i=1}^{\ell} Pr(y^\ell = y_c^\ell | x^\ell)$$

$$\prod_{i=1}^{\ell} Pr(y_c^i, x^i)$$

↳ Prob. of entire training data

$$= \prod_{i=1}^{\ell} Pr(y_c^i | x_6^i) \, Pr(x^i)$$

ignore

- Find *argmax* to give the best parameter settings.

$$\underset{W}{\text{argmax}} \prod_{\ell} Pr(y^\ell = y_c^\ell | x^\ell)$$

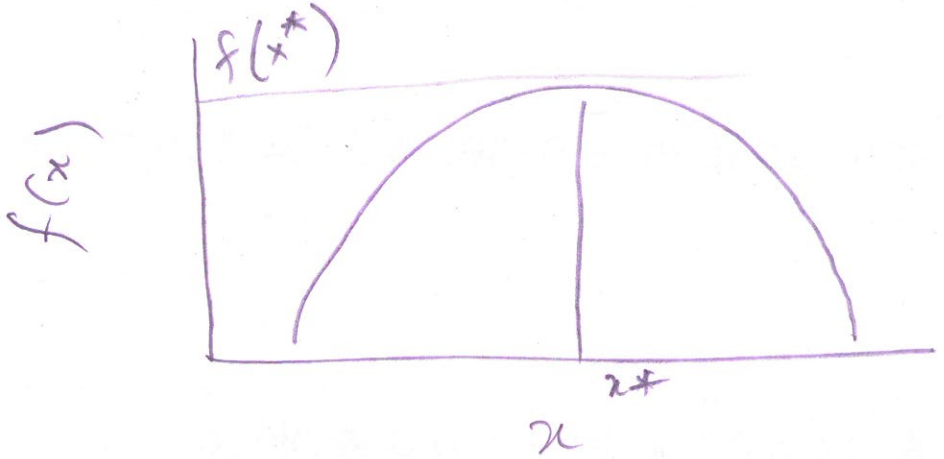
$$= \prod_{\ell \in \text{spam}} h_W(x^\ell) \prod_{\ell \in \text{ham}} (1 - h_W(x^\ell))$$

Optimization: Finding the max (or min) of a function

- How to find the *max* of our objective?

$$obj(\underline{w}) = \prod_{l \in S} h_w(x^l) \cdot \prod_{l \in H} (1 - h_w(x^l))$$

- How do you find the max of a function f(x)?



Take derivative

$f'(x)$

set it to zero

& solve for x

$f'(x) = 0$

Optimization: Finding the max (or min) of a function

- Let's compute the derivative of our objective and see if we can solve for W.

$$\text{argmax}_W \; Obj(W) = \prod_l Pr(y^l = y_c^l \mid x^l) = h_W$$

- Let's make our lives simpler by using a log-likelihood objective instead of likelihood.
  - Taking log leaves the argmax unchanged.

$$\text{argmax}_W \sum_l \ln \{ Pr(y^l = y_c^l \mid x^l) \}$$

  - Rewrite the summation to separate cases w/ $Y^l = 1$ and $Y^l = 0$

$$\text{argmax}_W \sum_l y_c^l \ln Pr(y^l = 1 \mid x^l) + (1 - y_c^l) \ln Pr(y^l = 0 \mid x^l)$$

  - Rearrange the terms to collect y_c terms in one place

$$\text{argmax}_W \sum y_c^l \ln \left\{ \frac{Pr(y^l = 1 \mid x^l)}{Pr(y^l = 0 \mid x^l)} \right\} + \ln Pr(y^l = 0 \mid x^l)$$

Issues: Generalization and Overfitting

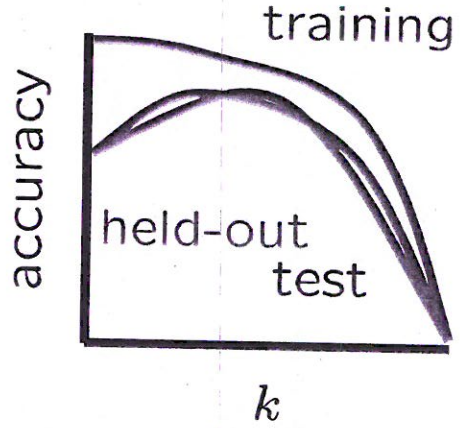
- Relative frequency parameters will overfit the training data!
  - Unlikely that every occurrence of "minute" is 100% spam
  - Unlikely that every occurrence of "seriously" is 100% ham
  - What about all the words that don't occur in the training set at all?
  - In general, we can't go around giving unseen events zero probability

- As an extreme case, imagine using the entire email as the only feature
  - Would get the training data perfect (if deterministic labeling)
  - Wouldn't *generalize* at all
  - Just making the bag-of-words assumption gives us some generalization, but isn't enough

- To generalize better: we need to smooth or regularize the estimates
  - Laplace's estimate
    - Pretend you saw every outcome k extra times

$$P_{LAP,k}(x) = \frac{c(x) + k}{N + k|X|}$$

    - What's Laplace with k = 0?
    - k is the strength of the prior
- Two kinds of unknowns, parameters and hyperparameters.



- Estimate probabilities from training data.
- Choose k based on held-out performance.

Issues: Independence assumptions

- Need more features– words aren't enough!
  - All CAPS
  - email domain
- Can add these information sources as new variables in the NB model
  - But The features are not independant.
  - Example:
    - "New" and "York"
    - Knowing "York" tells me New occurred before.
    - If you use P("New"|C) and P("York"|C) we are double counting.
- There are classifiers which let you easily add arbitrary features more easily.
  - We'll see one such next.

Logistic Regression Recipe: For binary classification

- Let's consider a linear combination function. $X = x_1, x_2, \ldots x_n$
  - Something that simply does a weighted combination of the feature values.

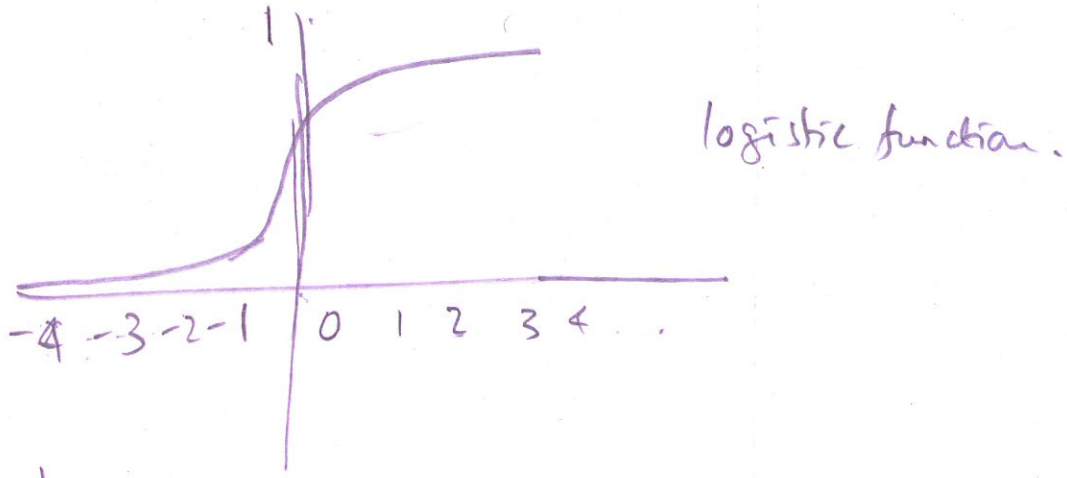
suppose $h_W(x) = W^T x = \sum_{i=1}^{n} w_i x_i$

- Does this form satisfy our constraints?

$0 \leq h_W(x) \leq 1$ $\quad w_i \in \mathbb{R} \;\; \forall \; w_i \in 1$

- What is the fix?

$$h_W(x) = \frac{e^{W^T x}}{1 + e^{W^T x}}$$



logistic function.

if $h_W(x) < 0.5 \Rightarrow$ ham
$h_W(x) \geq 0.5 \Rightarrow$ spam.

Optimization: Finding the max (or min) of a function

- Let's compute the derivative of our objective and see if we can solve for W.

- Let's make our lives simpler by using a log-likelihood objective instead of likelihood.

  - Taking log leaves the argmax unchanged. → L(W)

$$\underset{W}{\operatorname{argmax}} \sum_l \ln P(Y^l = y_c^l | X^l) \longrightarrow \text{log-likelihood objative.}$$

  - Rewrite the summation to separate cases w/ $Y^l = 1$ and $Y^l = 0$

$$\underset{W}{\operatorname{argmax}} \sum_l y_c^l \ln \left\{ P(Y^l = 1 | X^l) \right\} + (1 - y_c^l) \ln \left\{ P(Y^l = 0 | X^l) \right\}$$

  - Rearrange the terms to collect y_c terms in one place

$$\underset{W}{\operatorname{argmax}} \sum_l y_c^l \ln \frac{P(Y^l = 1 | X^l)}{P(Y^l = 0 | X^l)} + \ln P(Y^l = 0 | X^l)$$

Optimization: Find the max of the log-likelihood function.

$h_W(x) = \frac{e^{W^T X}}{1+e^{W^T X}}$ $\quad P(y^l=0|x^l) = 1 - h_W(x^l) = 1 - \frac{e^{W^T X}}{1+e^{W^T X}} = \frac{1}{1+e^{W^T X}}$

- Doing a bit of arithmetic will get you

$$\underset{W}{\operatorname{argmax}} \sum_l y_c^l \ln \frac{P(Y^l=1|X^l)}{P(Y^l=0|X^l)} + \ln P(Y^l=0|X^l)$$

$W^T X = z$

$y_c^l \ln(e^z) = y_c^l z$ $\quad \frac{e^z}{1+e^z} \Big/ \frac{1}{1+e^z}$

$$W^T X = \sum_{i=1}^{n} w_i X_i \qquad e^z \qquad \frac{1}{1+e^{W^T X}}$$

- Substituting back

$$\underset{W}{\operatorname{argmax}} \sum y_c^l W^T X + \ln\left(\frac{1}{1+e^{W^T X}}\right)$$

- Using ln(1/a) = - ln (a) →

$$\underset{W}{\operatorname{argmax}} \sum y_c^l W^T X - \ln\left(1+e^{W^T X}\right)$$

Optimization: Find the max of the log-likelihood function.

- Doing a bit of arithmetic will get you

sub h_W

$$\underset{W}{\operatorname{argmax}} \sum_l y_c^l \ln \frac{P(Y^l = 1|X^l)}{P(Y^l = 0|X^l)} + \ln P(Y^l = 0|X^l)$$

$$W^T X = \sum_{i=1}^{n} w_i X_i \qquad \frac{1}{1 + e^{W^T X}}$$

- Substituting back

$$\underset{W}{\operatorname{argmax}} \sum_l y_c^l \left(W^T X\right) + \ln \frac{1}{1 + e^{W^T X}}$$

- Using ln(1/a) = - ln (a) ➔

$$L(W) = \sum_l y_c^l \left(W^T X\right) - \ln\left(1 + e^{W^T X}\right)$$

argmax $L(W)$ → $\nabla L(W) = 0$
W → solve for W

Optimization: Find the max of the log-likelihood function.

- Let's take the gradient of this function w.r.t to W and set it to zero and solve for W.

$$L(W) = \sum_l y_c^l \left(W^T X\right) - \ln\left(1 + e^{W^T X}\right)$$

$W = \langle w_1, w_2 \cdots, w_n \rangle$

↑ wt of f_1 ↘ wt of f_2 ...

- The gradient is the set of partial derivatives w.r.t each W_i

$$\nabla L(W) = \left(\frac{\partial L(W)}{\partial W_1}, \frac{\partial L(W)}{\partial W_2}, \ldots, \frac{\partial L(W)}{\partial W_n}\right)$$

$W^T X = \sum_{j=1}^{n} w_j x_j$

$$\frac{\partial L(W)}{\partial W_i} = \sum_l \frac{\partial \left(y_c^l (W^T x)\right)}{\partial W_i} - \frac{\partial}{\partial W_i}\left(\ln\left(1 + e^{W^T x}\right)\right)$$

- Set this to zero. Can you solve for W_i?

$$\frac{\partial L(W)}{\partial W_i} = \sum_l y_c^l x_i - \frac{1}{1 + e^{W^T x}} \cdot e^{W^T x} \cdot x_i = 0 \quad \text{Solve for } W.$$

$y_c^l \frac{\partial W_i x_i}{\partial W_i}$

$\frac{1}{(1+e^{W^T x})} \cdot \frac{\partial (1 + e^{W^T x})}{\partial W_i}$

$\frac{1}{1+e^{W^T x}} \cdot e^{W^T x} x_i$

Optimization: Find the max of the log-likelihood function.

- Let's take the gradient of this function w.r.t to W

$$L(W) = \sum_l y_c^l \left(W^T X\right) - \ln\left(1 + e^{W^T X}\right)$$

$$\frac{d \ln(f(x))}{dx} = \frac{1}{f(x)} \frac{df(x)}{dx}$$

- This is the set of partial derivatives w.r.t each W_i

$$\nabla L(W) = \left(\frac{\partial L(W)}{\partial W_1}, \frac{\partial L(W)}{\partial W_2}, \ldots, \frac{\partial L(W)}{\partial W_n}\right)$$

$$\frac{\partial}{\partial W_i} \ln\left(1 + e^{W^T x}\right) = \frac{1}{1 + e^{W^T x}} \cdot \frac{\partial}{\partial x} e^{W^T x} \cdot \frac{\partial W^T x}{\partial x}$$

$$= \frac{1}{1 + e^{W^T x}} \cdot e^{W^T x} \cdot X_i$$

$$= \frac{e^{W^T x}}{1 + e^{W^T x}} \cdot X_i$$

- Where,

$$\frac{\partial L(W)}{\partial W_i} = \sum_l y_c^l X_i - \left(\frac{1}{1 + e^{W^T X}}\right)\left(e^{W^T X}\right) X_i$$

- You can set this to zero but you cannot solve for W_i because of the non-linearity!

Not to worry! Gradient ~~Descent~~ (Ascent) will save us.

Numerical method for the finding the maximum (and the arg-max).

$$\underset{W}{\operatorname{argmax}}\ L(W)$$

Idea:

Start with W ← <0, 0, …, 0>
Find an update to W that will increase L(W)
Iterate until some stopping criterion is met.

ie) $L(W^0) < L(W^1) < L(W^2) \leq \cdots$

Under some conditions this procedure will find the maximum

Intuition:

Gradient of a function at any point = slope of tangent plane at the point

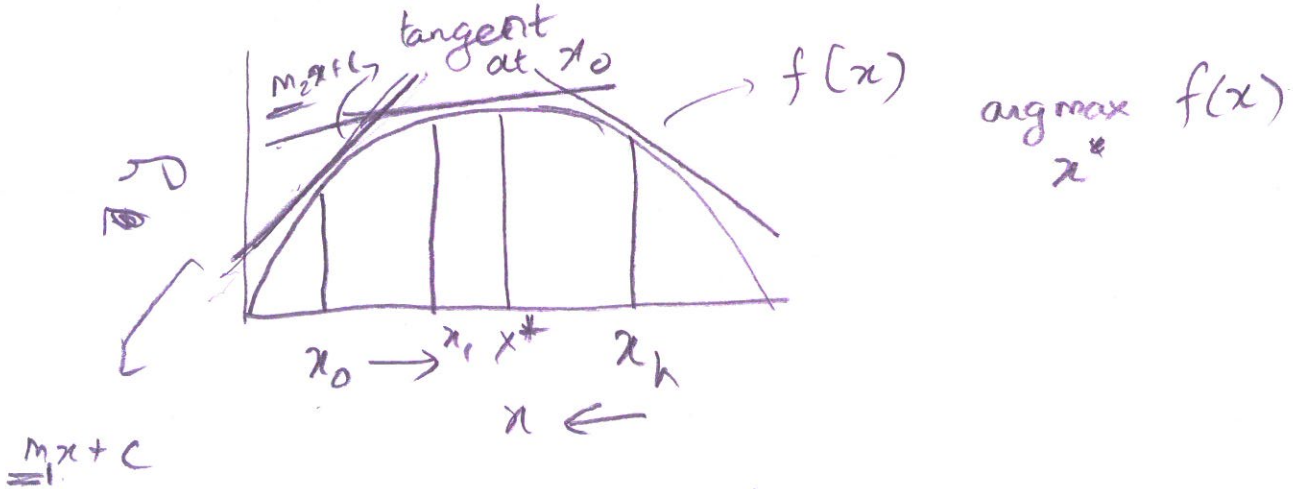
The slope provides the direction in which the function increases most
Take a small step in the gradient direction.

Not to worry! Gradient Descent (Ascent) will save us.

Gradient Ascent Intuition:

Gradient of a function at any point = slope of tangent plane at the point

- The slope provides the direction in which the function increases the most.
- Take a small step in the gradient direction.



$\arg\max_{x^*} f(x)$

$$x_0 \leftarrow x_0 + \eta \cdot \nabla f(x_0)$$

Gradient Ascent: Update rule

- Rule

$$W^{(t+1)} \leftarrow W^{(t)} + \eta \cdot \nabla L(W^{(t)})$$

- Recall the partial derivative is:

$$\frac{\partial L(W)}{\partial W_i} = \sum_l y_l^l x_i - \left[\frac{1}{1+e^{W^T x}} \cdot e^{W^T x} \right] x_i$$

$W^{(t)T}$

$$h_W(x) = P_\theta(y=1 \mid x)$$

- Two things to note:
  - Current estimate

$$\frac{\partial L(W)}{\partial W_i} = \sum_l y_l^l x_i - P_\theta(y=1 \mid x) \cdot x_i$$

↳ curr. estimate given $W^{(t)}$

  - Error term

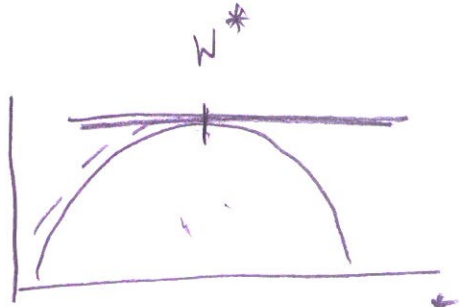
$$= \sum_l x_i \left(y_l^l - P_\theta(y=1 \mid x) \right)$$

↑ true label.

- Update rule per W_i :

$$w_i^{(t+1)} \leftarrow w_i^{(t)} + \eta \sum_l x_i^l \left(y_l^l - P_\theta(y=1 \mid x; W^{(t)}) \right)$$

Gradient Ascent: Termination, Convergence, and Alternatives



1) When to stop updating?

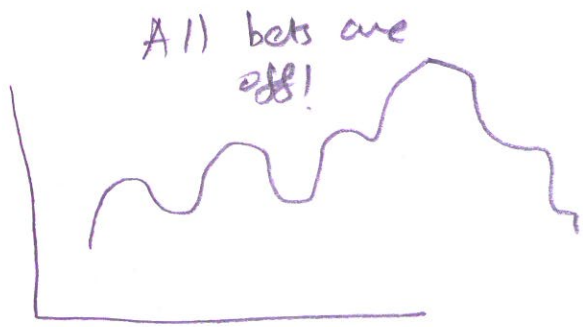
> Stop when w doesn't change.

↳ w will converge b/c at w^*, $w^{(t)} = w^*$

>

2) Convergence Guarantee

> Does it converge to the best w?



If $L(w)$ is concave, you will converge to global maximum.

3) Alternatives

> Gradient-based optimization methods

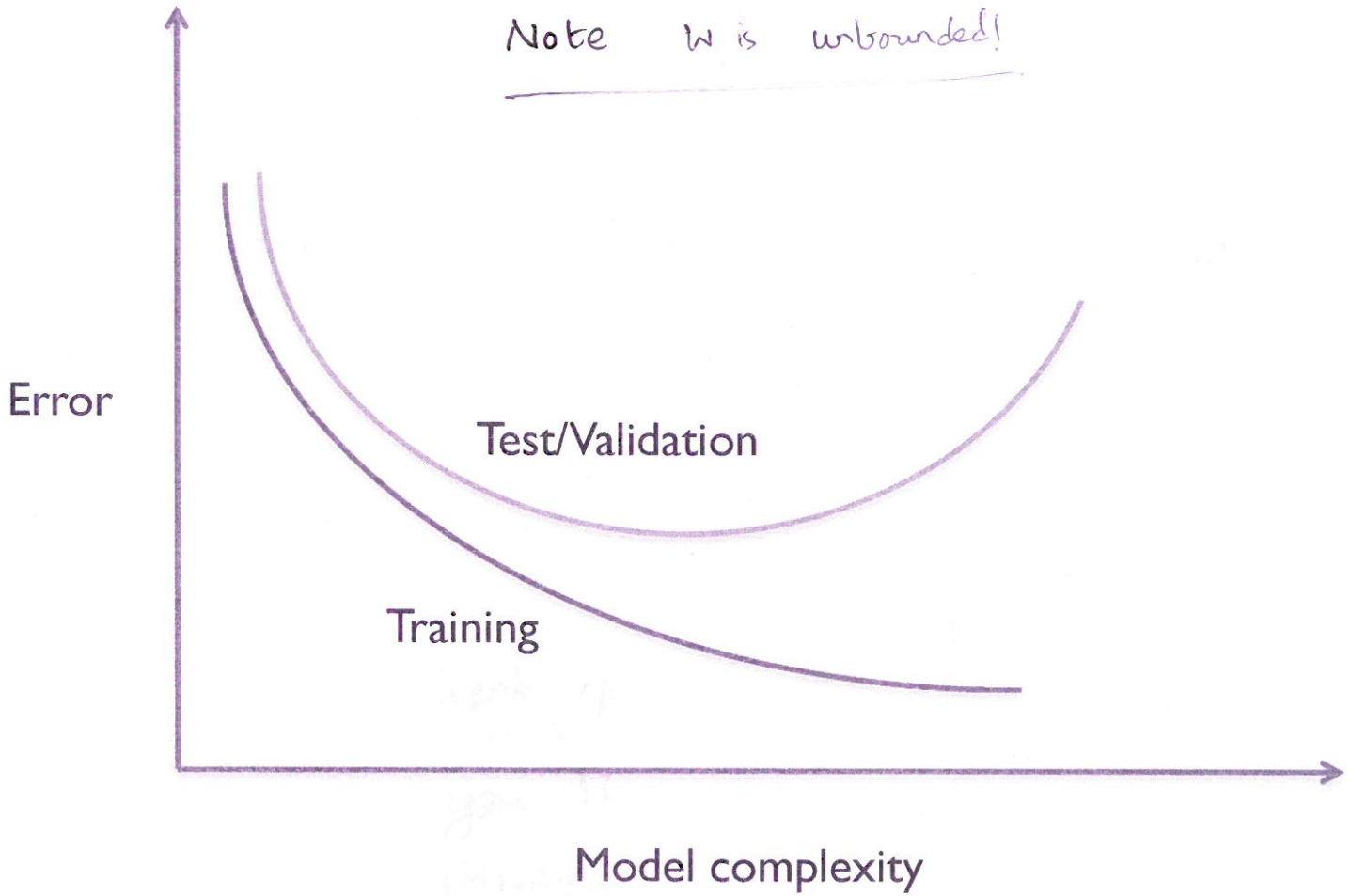
BFGS, L-BFGS

> Provide the gradient form & these methods give you the w^*

Regularization

Over-fitting

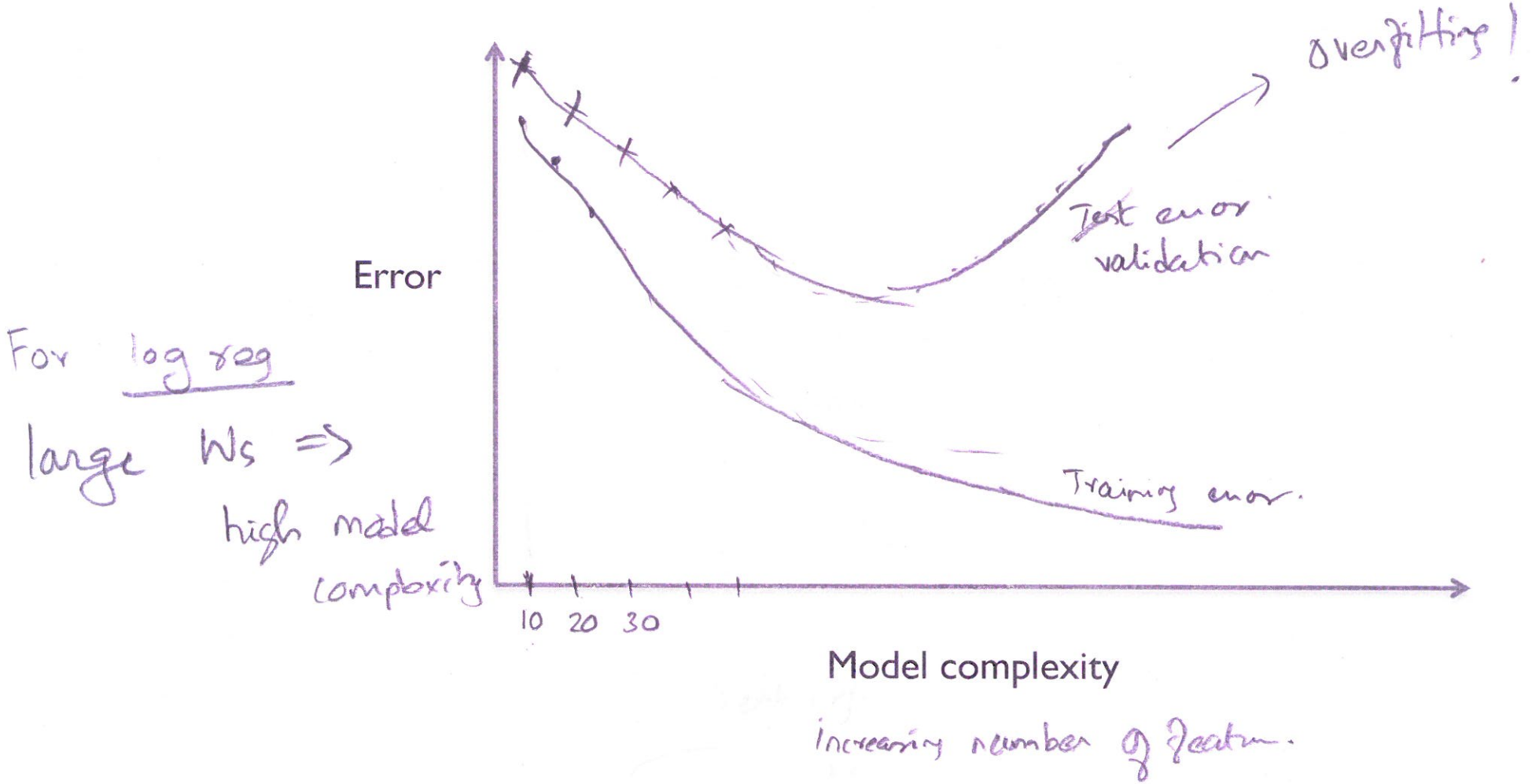
Training error decreases but not test error!



Overfitting and Regularization

Over-fitting: As you add more features i.e., parameters to your model

- The complexity of the model increases.
- The error decreases in training because you have more freedom to fit function.
- At the same time you might be *over fitting* to some noise
- This means your performance on unseen examples will suffer.



For log reg

large Ws => high model complexity

Regularization

> Issue: Over fitting results in large W values.

w_i^2

$2w_i$

> Fix? To insist that W be small.

(ie) $\|W\| \to 0$

$\|W\| = \sqrt{w_1^2 + w_2^2 + \dots + w_n^2}$

$\|W\|^2$

$$\underset{W}{\operatorname{argmax}} \; L(W) - \frac{\lambda}{2}\|W\|^2$$

↳ L2 reg.

Take gradient

> Update rule with regularization

alt L1 reg

$$w_i^{(t+1)} \leftarrow W_i^{(t)} + \eta \sum x_i^l \left(y_c^l - Pr(y=1 \mid x^l; W^{(t)})\right) - \eta \lambda W_i^{(t)}$$

↑ Parameters / hyper parameters.

Regularization

Over fitting results in large W values.

Regularization aims to enforce sparsity. i.e., small W

Add a penalty term to the Log-likelihood function

$$\underset{W}{\operatorname{argmax}} \; L(W) - \frac{\lambda}{2}||W||^2$$

Update rule with regularization parameter

$$w_i^{t+1} \leftarrow w_i^t + \eta \sum_l X_i(y_c^l - P(Y = 1|X; W^t)) - \eta\lambda w_i^t$$

Extending to multiple classes

> Binary: $Pr(y|x) \approx h_w(x)$

$\approx h_w(x) \leftarrow Pr(y=1|x)$

& $Pr(y=0|x)$

$= 1 - h_w(x)$

> Multiple classes.: $C = \{c_1, c_2, \ldots c_n\}$

$$Pr(y = c | x = x) = \frac{\exp\left(\sum w_i f_i(c, x)\right)}{\sum_{c_j \in C} \exp\left(\sum w_i f_i(c_j, x)\right)}$$

Feature functions: $f_i(c, x) \rightarrow$ is a feature that is present in x that is related to c.

e.g. $f_{32}(\overset{c}{\text{sport}}, x) = \begin{cases} 1 & \text{if "match"} \in x \wedge c = \text{sport} \\ 0 & \text{otherwise.} \end{cases}$

Logistic Regression Summary

- Discriminative version of Naïve Bayes
  - Models the class conditional directly.
- Often the first choice when the feature set is not too large.
  - Especially for binary classification.
- Easily extends to multiple classes
  - Maximum entropy classifier
- Useful references
  - Generative and Discriminative Classifiers (Chapter 1), Machine Learning, Tom Mitchell.
  - On Discriminative vs. Generative Classifiers: A comparison of Logistic Regression and Naïve Bayes, Ng and Jordan, NIPS 2002.
  - Andrew Ng's Coursera Lectures on Logistic Regression
    https://class.coursera.org/ml-003/lecture

] → Both are linear models that are identical when Train → ∞

→ log reg >> NB if when training data is larger.

→