Statistical NLP
Spring 2011

Lecture 9: Word Alignment II

Dan Klein – UC Berkeley

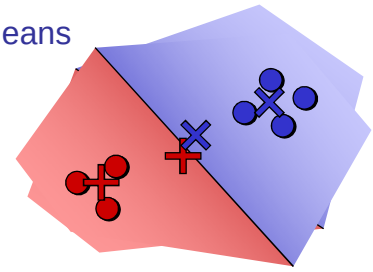
Learning with EM

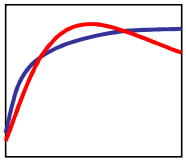
- Hard EM: alternate between

  E-step: Find best "completions" Y for fixed θ

  M-step: Find best parameters θ for fixed Y

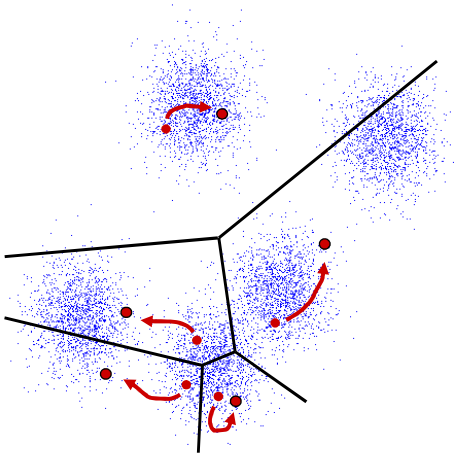
- Example: K-Means





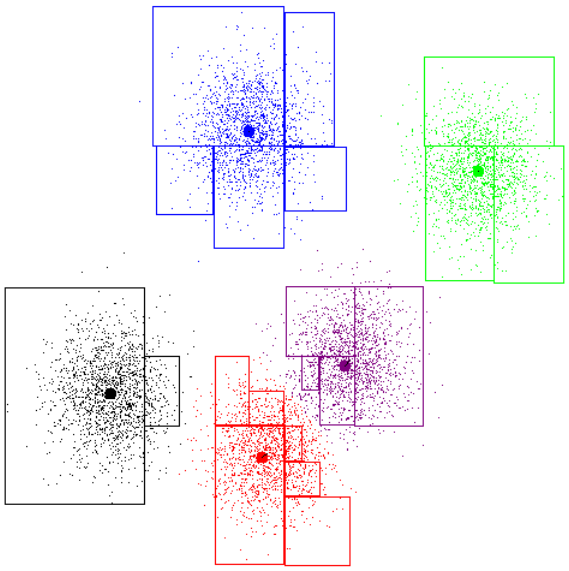
K-Means

- An iterative clustering algorithm
  - Pick K random points as cluster centers (means)
  - Alternate:
    - Assign data instances to closest mean
    - Assign each mean to the average of its assigned points
  - Stop when no points' assignments change



3

K-Means Example



4

K-Means Getting Stuck

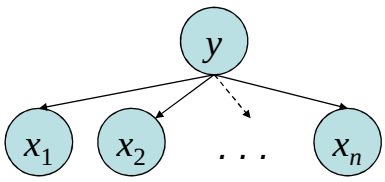
- A local optimum:

5

Naïve-Bayes Models

- Model: pick a topic, then generate a document using a language model for that topic.
- Naïve-Bayes assumption: all words are independent given the topic.

$$P(y, x_1, \ldots x_n) = P(y) \prod_i P(x_i|y)$$



Hard EM for Naïve-Bayes

- Procedure: (1) we calculate best completions:

$$y^* = \arg\max_y \mathsf{P}(y) \prod_i \mathsf{P}(x_i|y)$$

- (2) compute relevant counts from the completed data:

$$c(w, y) = \sum_{x \in D} \sum_i [1(x_i = w, y^* = y)]$$

- (3) compute new parameters from these counts (divide)
- (4) repeat until convergence
- Can also do this when some docs are labeled

Hard EM: More Formally

- Hard EM: $\arg\max_{\theta, y} \mathsf{P}(y, \theta|x)$
- Improve completions

$$y^* = \arg\max_y \mathsf{P}(y, \theta^*|x) = \arg\max_y \mathsf{P}(y|x, \theta^*)$$

- Improve parameters

$$\theta^* = \arg\max_\theta \mathsf{P}(y^*, \theta|x) = \arg\max_\theta \mathsf{P}(\theta|x, y^*)$$

- Each step either does nothing or increases the objective

Soft EM for Naïve-Bayes

- Procedure: (1) calculate posteriors (soft completions):

$$\mathsf{P}(y|x) = \frac{\mathsf{P}(y)\prod_i \mathsf{P}(x_i|y)}{\sum_{y'} \mathsf{P}(y')\prod_i \mathsf{P}(x_i|y')}$$

- (2) compute expected counts under those posteriors:

$$c(w,y) = \sum_{x \in D} \mathsf{P}(y|x) \sum_i [1(x_i = w, y)]$$

- (3) compute new parameters from these counts (divide)
- (4) repeat until convergence

EM in General

- We'll use EM over and over again to fill in missing data
  - Convenience Scenario: we want P(x), including y just makes the model simpler (e.g. mixing weights for language models)
  - Induction Scenario: we actually want to know y (e.g. clustering)
  - NLP differs from much of statistics / machine learning in that we often want to interpret or use the induced variables (which is tricky at best)
- General approach: alternately update y and θ
  - E-step: compute posteriors P(y|x,θ)
    - This means scoring all completions with the current parameters
    - Usually, we do this implicitly with dynamic programming
  - M-step: fit θ to these completions
    - This is usually the easy part – treat the completions as (fractional) complete data
  - Initialization: start with some noisy labelings and the noise adjusts into patterns based on the data and the model
  - We'll see lots of examples in this course
- EM is only locally optimal (why?)

KL Divergence

KL measures how different two distributions p and q are.

Definition:

$$\mathrm{KL}(q||p) \stackrel{\text{def}}{=} \mathbb{E}_q \log \frac{q(\theta)}{p(\theta)}$$

An important property:

$\mathrm{KL}(q||p) \geq 0$ $\quad$ $\mathrm{KL}(q||p) = 0$ if and only if $q = p$

KL is asymmetric:

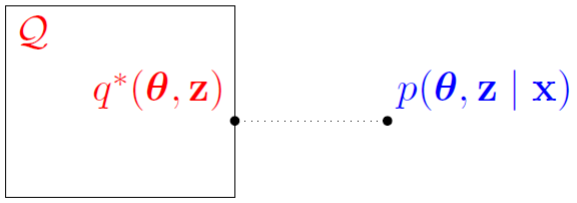
Assuming $\mathrm{KL}(q||p) < \infty$,

$$p(\theta) = 0 \Rightarrow q(\theta) = 0 \quad [q \text{“} \subset \text{”} p]$$

General Setup

- KL divergence to true posterior

$$q^* = \mathrm{argmin}_{q \in \mathcal{Q}} \mathrm{KL}(q(\boldsymbol{\theta}, \mathbf{z}) || p(\boldsymbol{\theta}, \mathbf{z} \mid \mathbf{x}))$$



Approximations

$q(\boldsymbol{\theta}, \mathbf{z}) = q(\boldsymbol{\theta})\, q(\mathbf{z})$

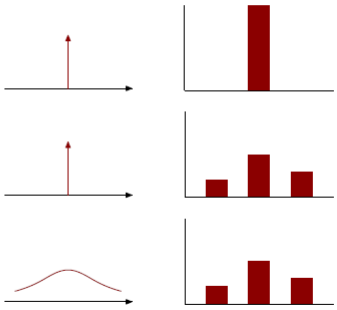
Hard EM

↓

EM

↓

mean-field



General Solution

$$q^* = \underset{q \in \mathcal{Q}}{\text{argmin}}\, \text{KL}(q(\boldsymbol{\theta}, \mathbf{z}) || p(\boldsymbol{\theta}, \mathbf{z} \mid \mathbf{x}))$$

Steps:

1. Formulate as an optimization problem (variational principle)
2. Relax the optimization problem (e.g., mean-field)
3. Solve using coordinate-wise descent

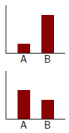
Example: Two-Mixture

$\phi_1 \sim \text{Dirichlet}(1, 1)$

$\phi_2 \sim \text{Dirichlet}(1, 1)$

$z_i \sim \text{Multinomial}(\frac{1}{2}, \frac{1}{2})$

$x_i \sim \text{Multinomial}^5(\phi_{z_i})$ (A) (A) (B) (B) (A)

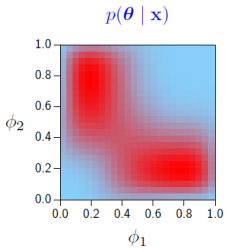


Observed data:

$x_1 =$ (A) (B) (B) (B) (B)
$x_2 =$ (A) (B) (B) (B) (B)
$x_3 =$ (B) (A) (A) (A) (A)

Unknown parameters:

$\boldsymbol{\theta} = (\phi_1, \phi_2)$

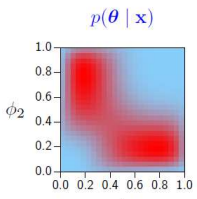


- True posterior $p(\boldsymbol{\theta} \mid \mathbf{x})$ has symmetries
- ϕ_1 explains x_1, x_2 and ϕ_2 explains x_3 in upper mode (or vice-versa in lower mode)
- The component explaining x_3 has higher uncertainty

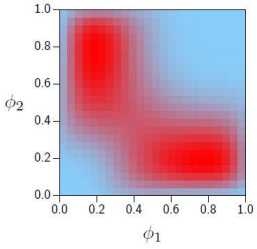
Example Posteriors

$$q^* \stackrel{\text{def}}{=} \operatorname*{argmin}_{q \in \mathcal{Q}_{\text{all}}} \text{KL}(q(\boldsymbol{\theta}) \,||\, p(\boldsymbol{\theta} \mid \mathbf{x}))$$

$\mathcal{Q}_{\text{all}} =$ all distributions

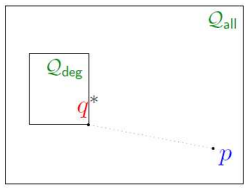


Optimal $q^*(\boldsymbol{\theta})$

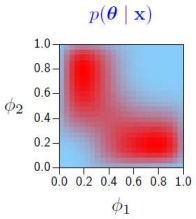


Approximate Posteriors

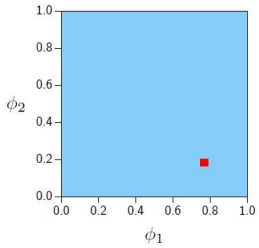
$$q^* \stackrel{\text{def}}{=} \operatorname*{argmin}_{q \in \mathcal{Q}_{\text{deg}}} \mathrm{KL}(q(\boldsymbol{\theta}) \,||\, p(\boldsymbol{\theta} \mid \mathbf{x}))$$



$$\mathcal{Q}_{\text{deg}} = \left\{ q : q(\boldsymbol{\theta}) = \delta_{\boldsymbol{\theta}^*}(\boldsymbol{\theta}) \right\}$$

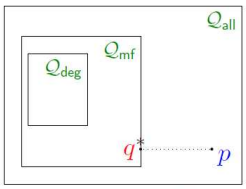


Degenerate $q^*(\boldsymbol{\theta})$

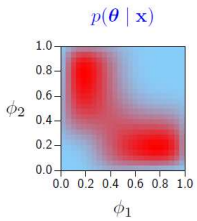


Approximate Posteriors

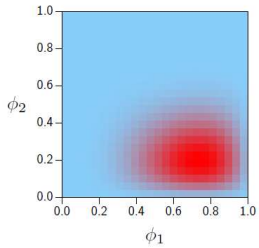
$$q^* \stackrel{\text{def}}{=} \operatorname*{argmin}_{q \in \mathcal{Q}_{\text{mf}}} \mathrm{KL}(q(\boldsymbol{\theta}) \,||\, p(\boldsymbol{\theta} \mid \mathbf{x}))$$



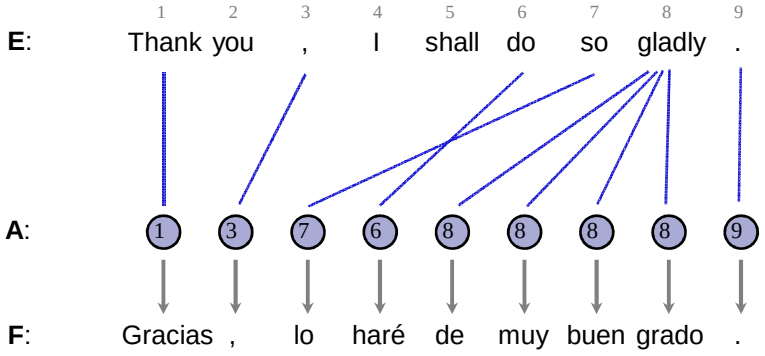
$$\mathcal{Q}_{\text{mf}} = \left\{ q : q(\boldsymbol{\theta}) = \prod_{i=1}^{n} q_i(\theta_i) \right\}$$



Mean-field $q^*(\boldsymbol{\theta})$



IBM Models 1/2

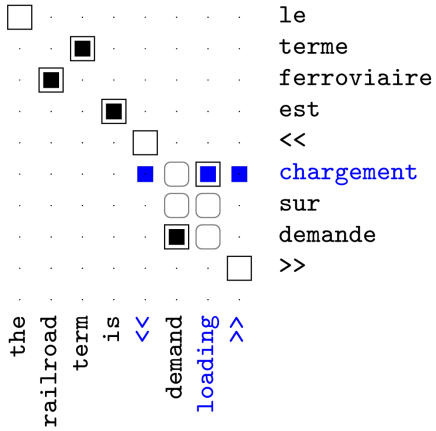


Model Parameters

Emissions: $P(F_1 = \text{Gracias} \mid E_{A_1} = \text{Thank})$ *Transitions*: $P(A_2 = 3)$

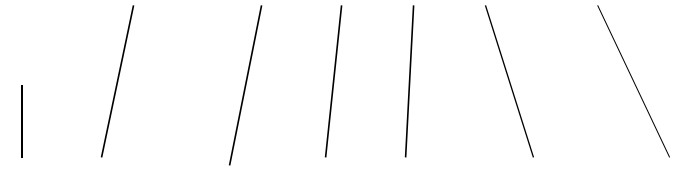
Problems with Model 1

- There's a reason they designed models 2-5!
- Problems: alignments jump around, align everything to rare words
- Experimental setup:
  - Training data: 1.1M sentences of French-English text, Canadian Hansards
  - Evaluation metric: alignment error Rate (AER)
  - Evaluation data: 447 hand-aligned sentences



Monotonic Translation

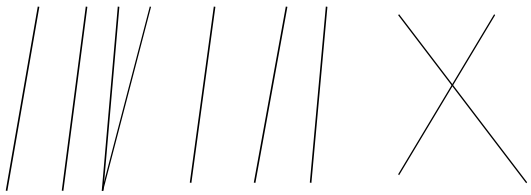
Japan shaken by two new quakes



Le Japon secoué par deux nouveaux séismes

Local Order Change

Japan is at the junction of four tectonic plates



Le Japon est au confluent de quatre plaques tectoniques

IBM Model 2

- Alignments tend to the diagonal (broadly at least)

$$P(f, a|e) = \prod_j P(a_j = i|j, I, J) P(f_j|e_i)$$

$$P(dist = i - j\frac{I}{J})$$

$$\frac{1}{Z} e^{-\alpha(i - j\frac{I}{J})}$$

- Other schemes for biasing alignments towards the diagonal:
  - Relative vs absolute alignment
  - Asymmetric distances
  - Learning a full multinomial over distances

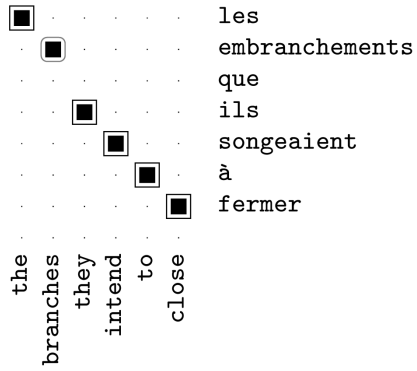
EM for Models 1/2

- Model 1 Parameters:
  Translation probabilities (1+2) $P(f_j|e_i)$
  Distortion parameters (2 only) $P(a_j = i|j, I, J)$
- Start with $P(f_j|e_i)$ uniform, including $P(f_j|null)$
- For each sentence:
  - For each French position j
    - Calculate posterior over English positions

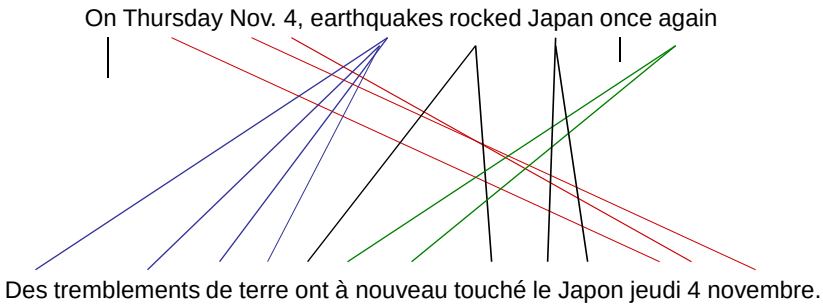
$$P(a_j = i|f, e) = \frac{P(a_j = i|j, I, J) P(f_j|e_i)}{\sum_{i'} P(a_j = i'|j, I, J) P(f_j|e_{i'})}$$

    - (or just use best single alignment)
    - Increment count of word f_j with word e_i by these amounts
    - Also re-estimate distortion probabilities for model 2
- Iterate until convergence

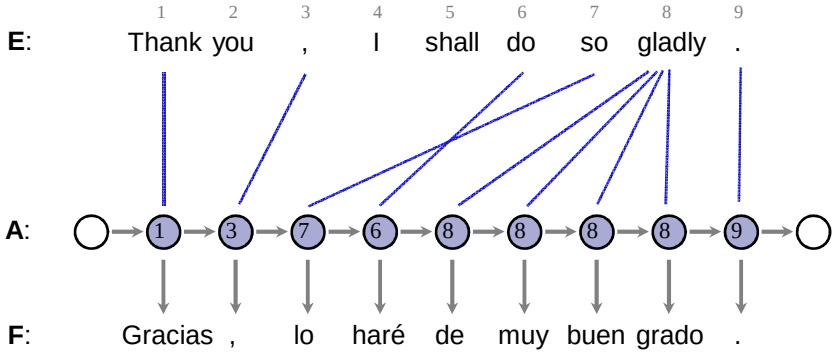
Example



Phrase Movement



The HMM Model



Model Parameters

Emissions: $P(F_1 = \text{Gracias} \mid E_{A_1} = \text{Thank})$ *Transitions*: $P(A_2 = 3 \mid A_1 = 1)$

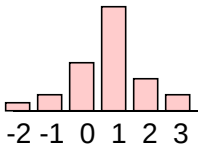
The HMM Model

- Model 2 preferred global monotonicity
- We want local monotonicity:
  - Most jumps are small
- HMM model (Vogel 96)

f	$t(f \mid e)$
nationale	0.469
national	0.418
nationaux	0.054
nationales	0.029

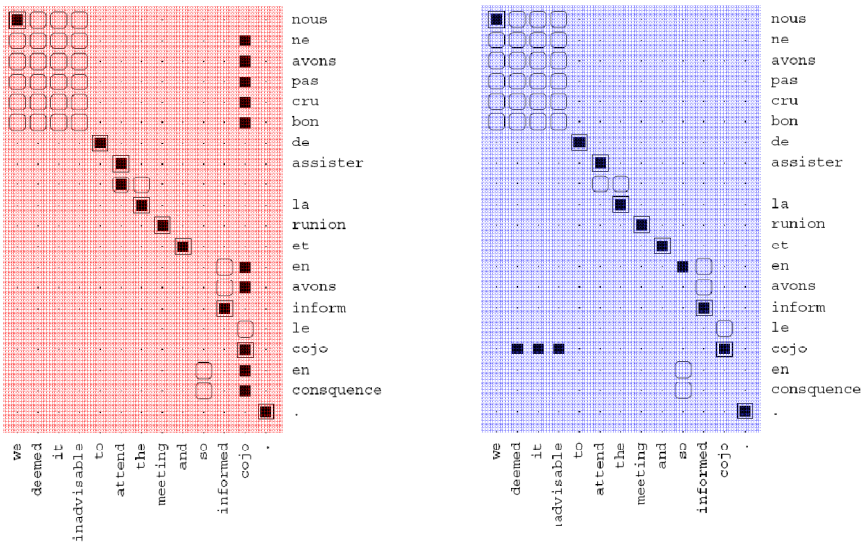
$$P(f, a|e) = \prod_j P(a_j|a_{j-1})P(f_j|e_i)$$

$P(a_j - a_{j-1})$



  - Re-estimate using the forward-backward algorithm
  - Handling nulls requires some care
- What are we still missing?

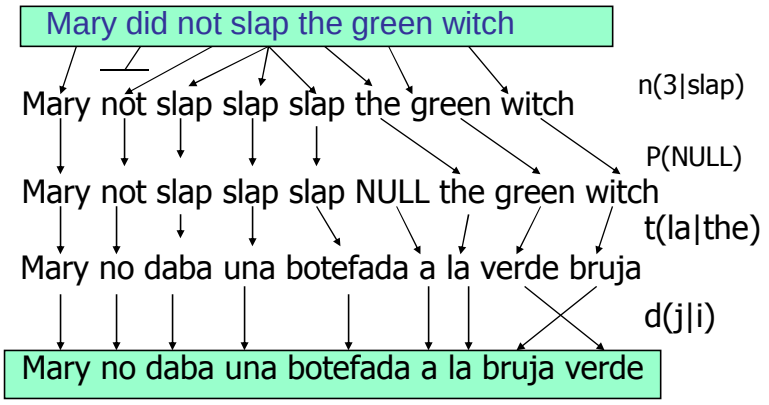
HMM Examples



AER for HMMs

Model	AER
Model 1 INT	19.5
HMM E→F	11.4
HMM F→E	10.8
HMM AND	7.1
HMM INT	4.7
GIZA M4 AND	6.9

IBM Models 3/4/5



[from Al-Onaizan and Knight, 1998]

Examples: Translation and Fertility

the

f	$t(f \mid e)$	ϕ	$n(\phi \mid e)$
le	0.497	1	0.746
la	0.207	0	0.254
les	0.155		
l′	0.086		
ce	0.018		
cette	0.011		

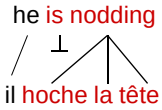
not

f	$t(f \mid e)$	ϕ	$n(\phi \mid e)$
ne	0.497	2	0.735
pas	0.442	0	0.154
non	0.029	1	0.107
rien	0.011		

farmers

f	$t(f \mid e)$	ϕ	$n(\phi \mid e)$
agriculteurs	0.442	2	0.731
les	0.418	1	0.228
cultivateurs	0.046	0	0.039
producteurs	0.021		

Example: Idioms



nodding

f	$t(f \mid e)$	ϕ	$n(\phi \mid e)$
signe	0.164	4	0.342
la	0.123	3	0.293
tête	0.097	2	0.167
oui	0.086	1	0.163
fait	0.073	0	0.023
que	0.073		
hoche	0.054		
hocher	0.048		
faire	0.030		
me	0.024		
approuve	0.019		
qui	0.019		
un	0.012		
faites	0.011		

Example: Morphology

should

f	$t(f \mid e)$	ϕ	$n(\phi \mid e)$
devrait	0.330	1	0.649
devraient	0.123	0	0.336
devrions	0.109	2	0.014
faudrait	0.073		
faut	0.058		
doit	0.058		
aurait	0.041		
doivent	0.024		
devons	0.017		
devrais	0.013		

Some Results

- [Och and Ney 03]

Model	Training scheme	0.5K	8K	128K	1.47M
Dice		50.9	43.4	39.6	38.9
Dice+C		46.3	37.6	35.0	34.0
Model 1	1^5	40.6	33.6	28.6	25.9
Model 2	$1^5 2^5$	46.7	29.3	22.0	19.5
HMM	$1^5 H^5$	26.3	23.3	15.0	10.8
Model 3	$1^5 2^5 3^3$	43.6	27.5	20.5	18.0
	$1^5 H^5 3^3$	27.5	22.5	16.6	13.2
Model 4	$1^5 2^5 3^3 4^3$	41.7	25.1	17.3	14.1
	$1^5 H^5 3^3 4^3$	26.1	20.2	13.1	9.4
	$1^5 H^5 4^3$	26.3	21.8	13.3	9.3
Model 5	$1^5 H^5 4^3 5^3$	26.5	21.5	13.7	9.6
	$1^5 H^5 3^3 4^3 5^3$	26.5	20.4	13.4	9.4
Model 6	$1^5 H^5 4^3 6^3$	26.0	21.6	12.8	8.8
	$1^5 H^5 3^3 4^3 6^3$	25.9	20.3	12.5	8.7