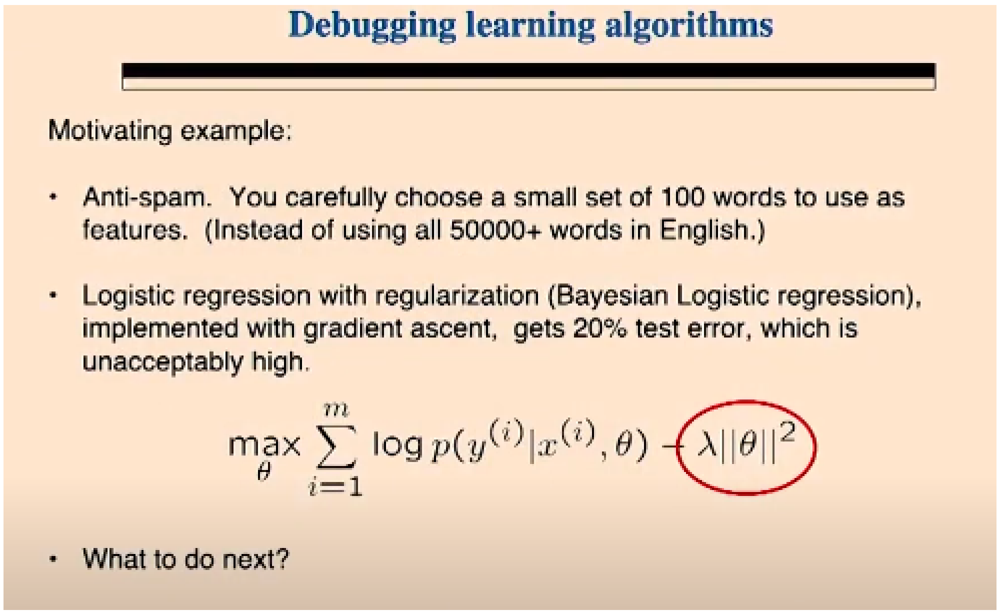
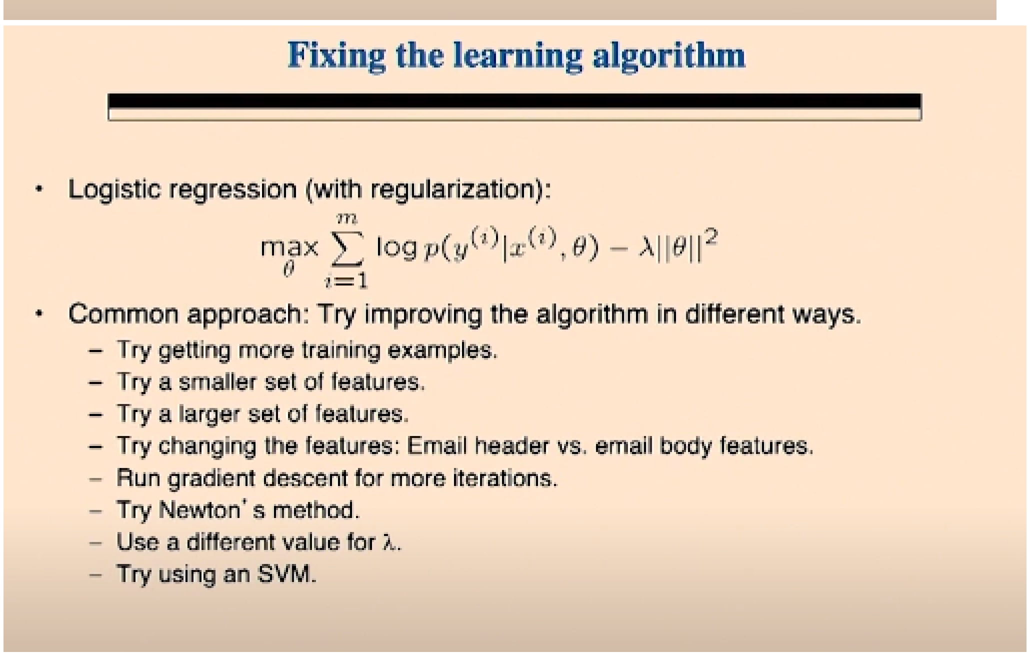
Lecture 13 - Debugging ML Models and Error Analysis | Stanford CS229: Machine Learning (Autumn 2018)-Make Written Notes

Notebook:	100-Paper-Notes
Created:	01/06/2020 02:50
Updated:	01/06/2020 17:58
Author:	mohammad.mohtashim76@gmail.com

- 

- 

- ## Diagnostic for bias vs. variance

  Better approach:
  - Run diagnostics to figure out what the problem is.
  - Fix whatever the problem is.

  Logistic regression's test error is 20% (unacceptably high).

  Suppose you suspect the problem is either:
  - Overfitting (high variance).
  - Too few features to classify spam (high bias).

  Diagnostic:
  - Variance: Training error will be much lower than test error.
  - Bias: Training error will also be high.

-

- ## More on bias vs. variance

  Typical learning curve for high variance:

  • Test error still decreasing as m increases. Suggests larger training set will help.
  • Large gap between training and test error.

  Andrew Y. Ng

- ## More on bias vs. variance

  Typical learning curve for high bias:

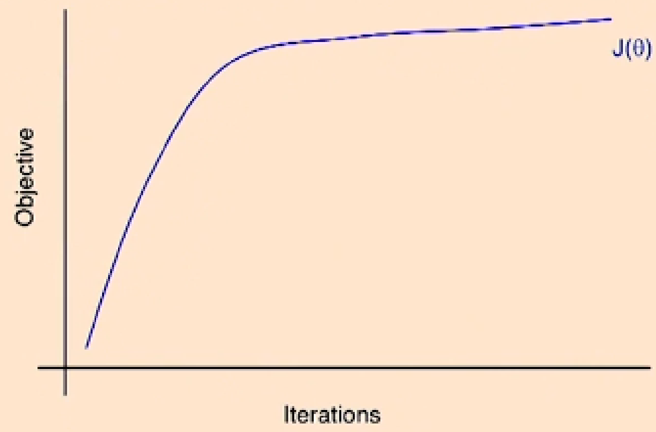
  • Even training error is unacceptably high.
  • Small gap between training and test error.

- ## Optimization algorithm diagnostics

  - Bias vs. variance is one common diagnostic.
  - For other problems, it's usually up to your own ingenuity to construct your own diagnostics to figure out what's wrong.
  - Another example:
    - Logistic regression gets 2% error on spam, and 2% error on non-spam. (Unacceptably high error on non-spam.)
    - SVM using a linear kernel gets 10% error on spam, and 0.01% error on non-spam. (Acceptable performance.)
    - But you want to use logistic regression, because of computational efficiency, etc.
  - What to do next?

More diagnostics

- Other common questions:
  - Is the algorithm (gradient ascent for logistic regression) converging?



More diagnostics

- Other common questions:
  - Is the algorithm (gradient ascent for logistic regression) converging?
  - Are you optimizing the right function?
  - I.e., what you care about:

$$a(\theta) = \sum_i w^{(i)} 1\{h_\theta(x^{(i)}) = y^{(i)}\}$$

(weights $w^{(i)}$ higher for non-spam than for spam).

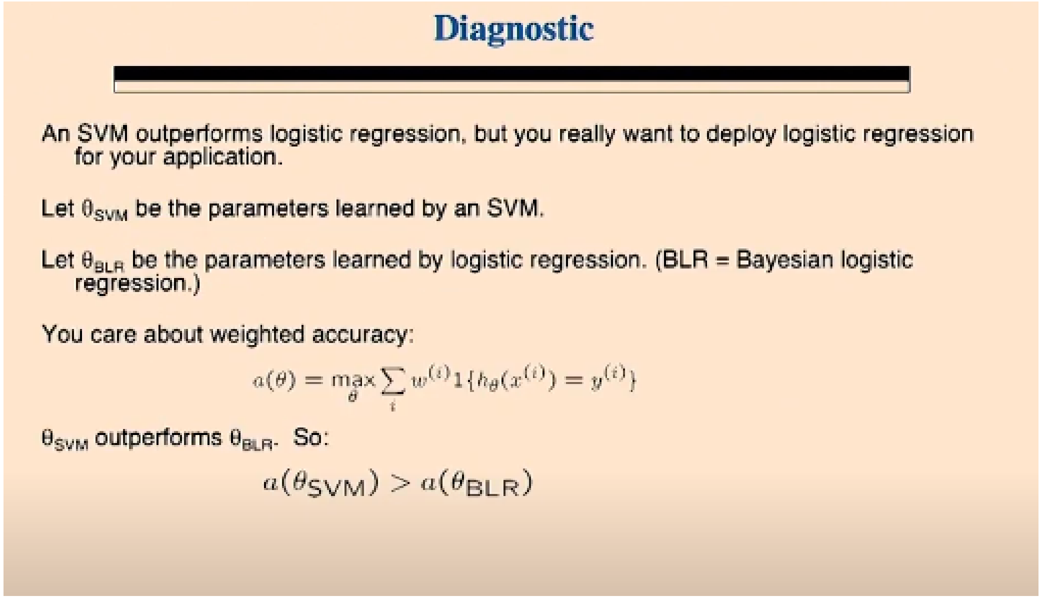
  - Logistic regression? Correct value for λ?

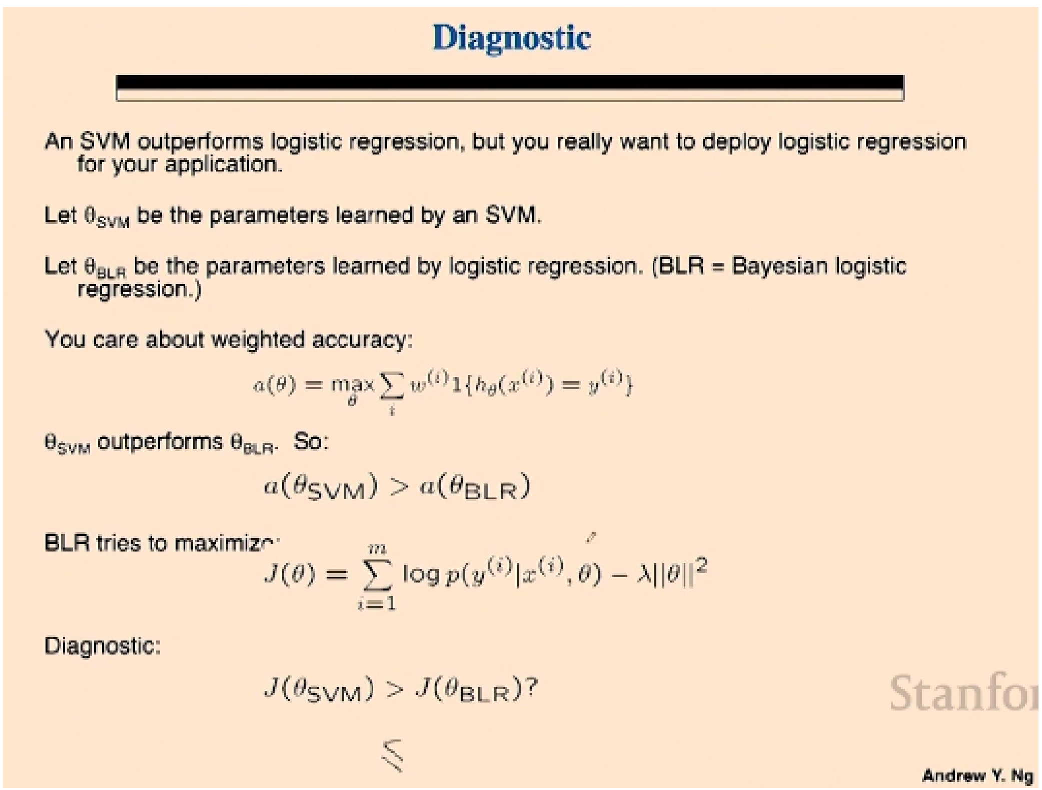
$$\max_\theta J(\theta) = \sum_{i=1}^{m} \log p(y^{(i)}|x^{(i)}, \theta) - \lambda ||\theta||^2$$

  - SVM? Correct value for C?

$$\min_{w,b} \quad ||w||^2 + C \textstyle\sum_{i=1}^{m} \xi_i$$

$$\text{s.t.} \quad y^{(i)}(w^T x^{(i)} - b) \geq 1 - \xi_i$$

- 

- 

- Very Important-New Insight-Please See it Future Mohtashim

- ## Two cases

  Case 1: $a(\theta_{\text{SVM}}) > a(\theta_{\text{BLR}})$
  $$J(\theta_{\text{SVM}}) > J(\theta_{\text{BLR}})$$

  But BLR was trying to maximize $J(\theta)$. This means that θ_{BLR} fails to maximize J, and the problem is with the convergence of the algorithm. Problem is with optimization algorithm.

  Case 2: $\boxed{a(\theta_{\text{SVM}}) > a(\theta_{\text{BLR}})}$
  $$J(\theta_{\text{SVM}}) \leq J(\theta_{\text{BLR}})$$

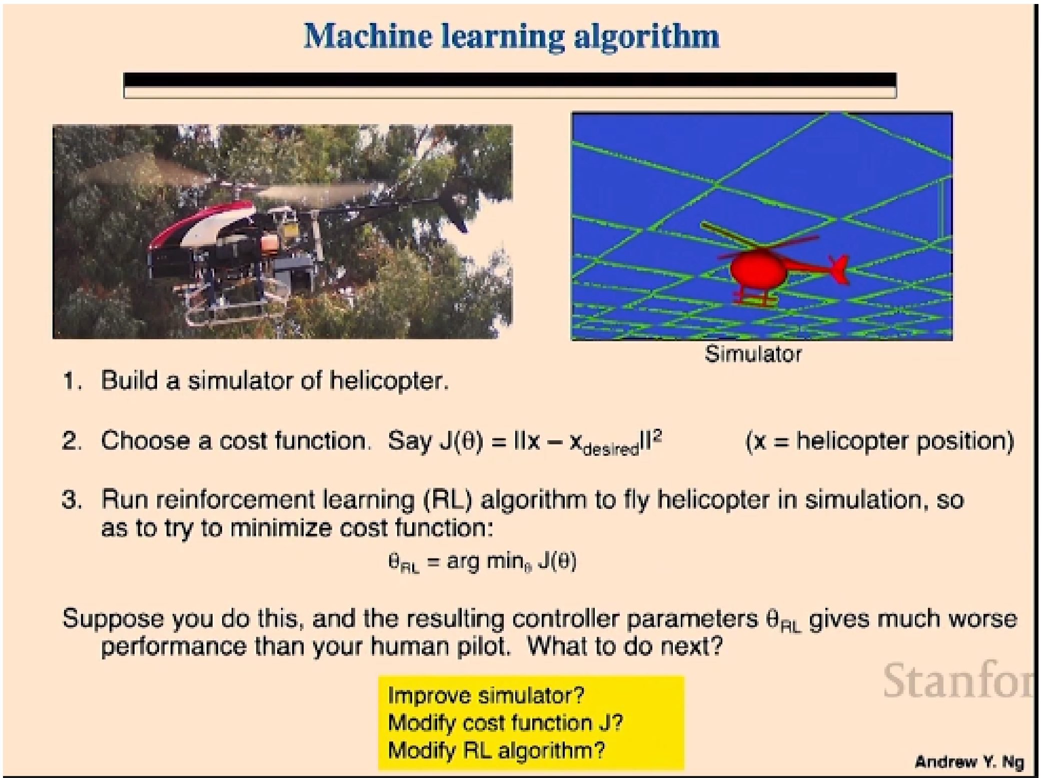
  This means that BLR succeeded at maximizing $J(\theta)$. But the SVM, which does worse on $J(\theta)$, actually does better on weighted accuracy $a(\theta)$.

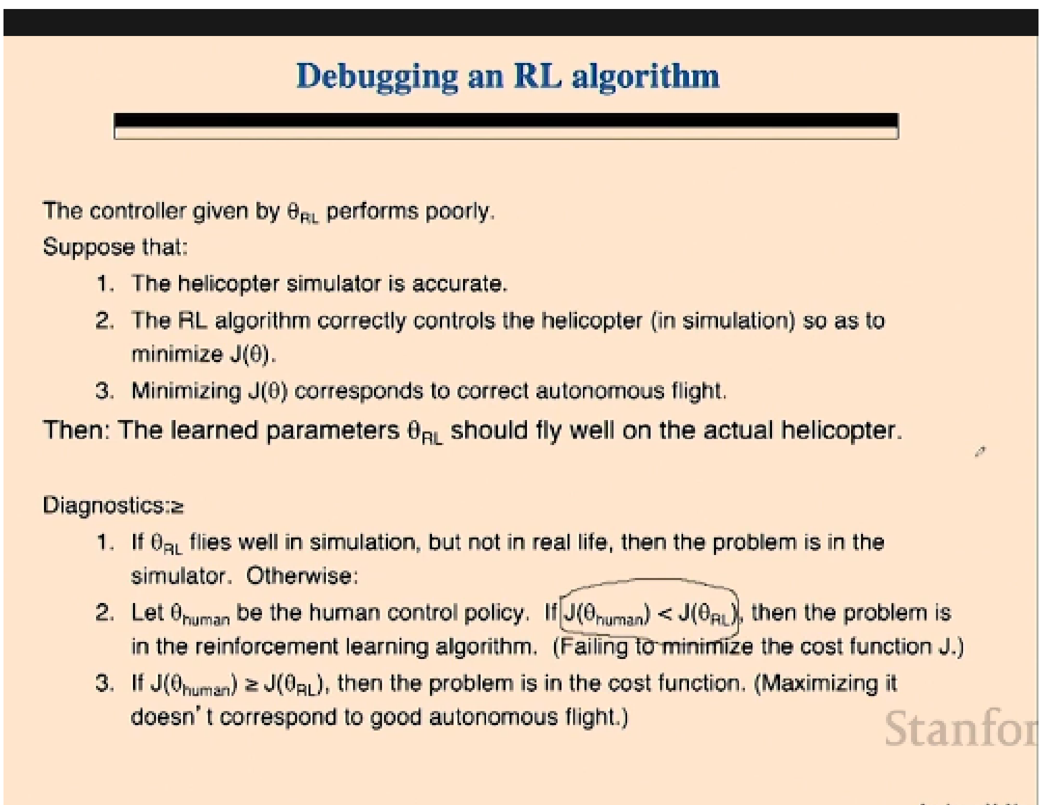
  This means that $J(\theta)$ is the wrong function to be maximizing, if you care about $a(\theta)$. Problem is with objective function of the maximization problem.

- ## The Stanford Autonomous Helicopter

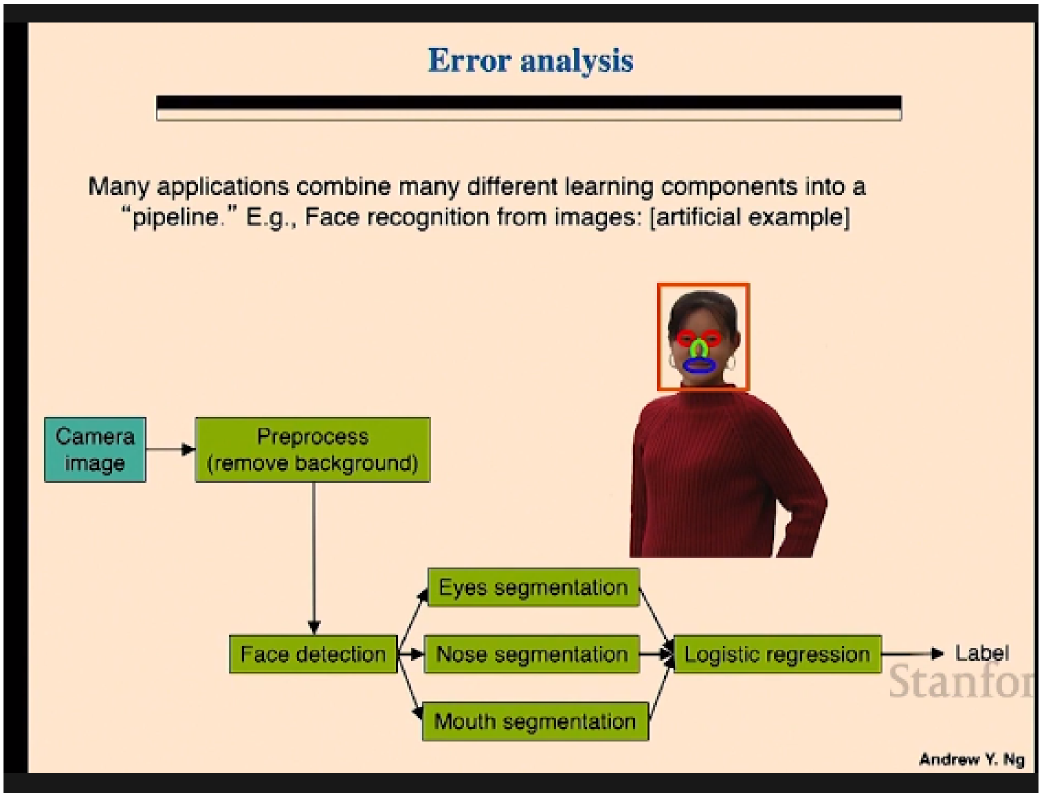
  

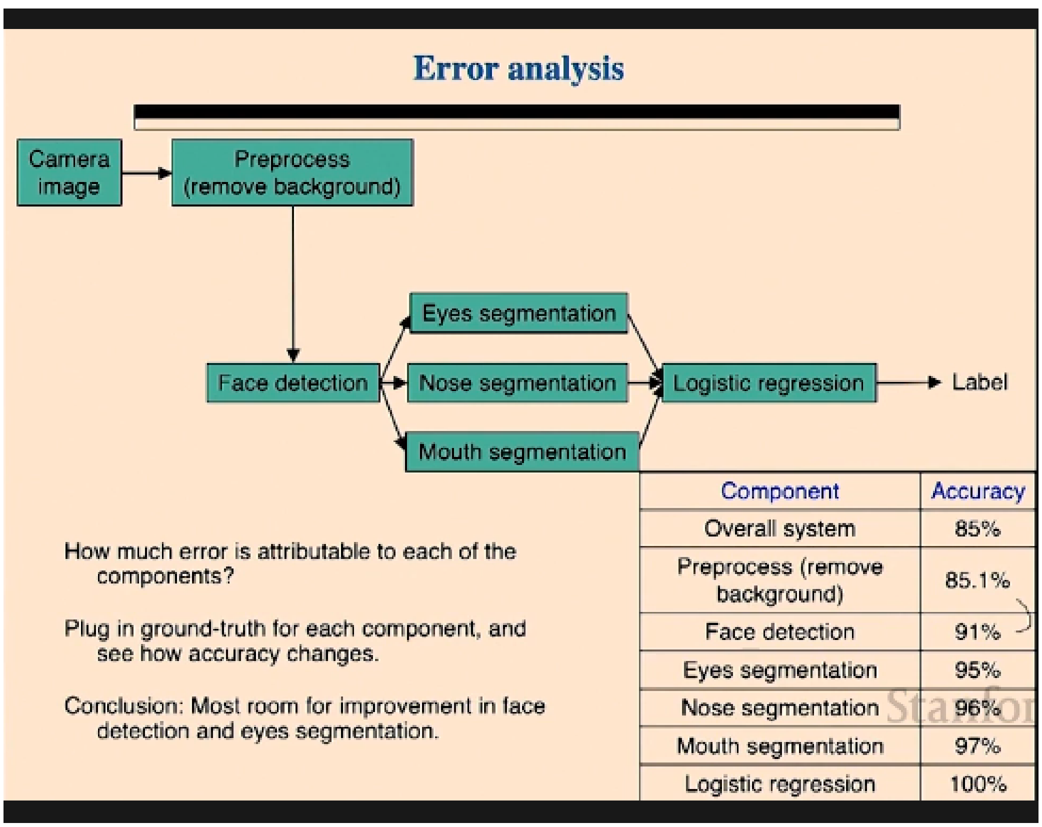
  Payload: 14 pounds
  Weight: 32 pounds

- 

- 

- Remember two things:
  - Bias/Variance Trade-off
  - Analyse whether the problem is in algorithm or objective that you are trying to achieve

- 



Component	Accuracy
Overall system	85%
Preprocess (remove background)	85.1%
Face detection	91%
Eyes segmentation	95%
Nose segmentation	96%
Mouth segmentation	97%
Logistic regression	100%

Ablative analysis

Simple logistic regression without any clever features get 94% performance.

Just what accounts for your improvement from 94 to 99.9%?

Ablative analysis: Remove components from your system one at a time, to see how it breaks.

Component	Accuracy	
Overall system	99.9%	
Spelling correction	99.0	
Sender host features	98.9%	
Email header features	98.9%	
Email text parser features	95%	
Javascript parser	94.5%	
Features from images	94.0%	[baseline]

Conclusion: The email text parser features account for most of the improvement.