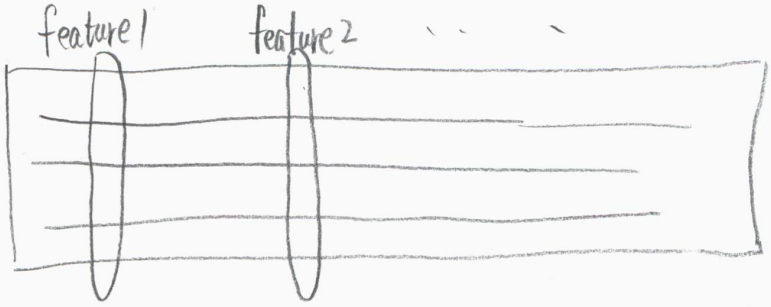
Akuma.txt (102 features)

① Score function

Suppose we have a partition T now. $T = \{t_1, t_2, \cdots, t_n\}$



$f(t^1) \leftarrow (Var1, Var2, \cdots, Var102)$

$f(t^2) \in$ ...

⋮

Score for cluster1: $S(t^1) = f(t^1)^T \times \Lambda$

where $f(t^1) \in R^{102}$, $\Lambda \in R^{102}$

Score for cluster2: $S(t^2) = f(t^2) \times \Lambda$

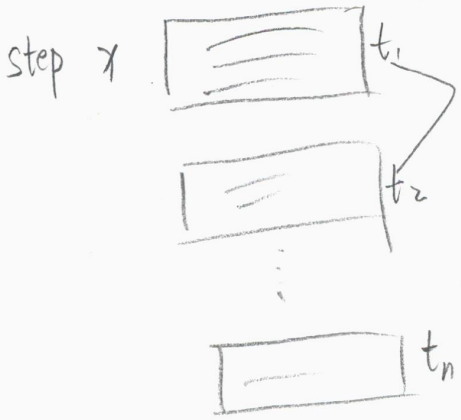
⋮

Score for partition T is $S(T) = S(t^1) + S(t^2) + \cdots + S(t^n)$

$$= \left[\sum_i f(t^i)^T\right] \times \Lambda$$

$S(T)$ smaller → Partition is better

$S^*(T)$ = the accuracy of T



step $x+1$

$$S(t_1 \cup t_2) + S(t_3) + \cdots S(t_n)$$

$\because S(t_1) + \cdots + S(t_n)$ is fixed

$\therefore$ We just calculate $\underline{S(t_1 \cup t_2) - S(t_1) - S(t_2)}$

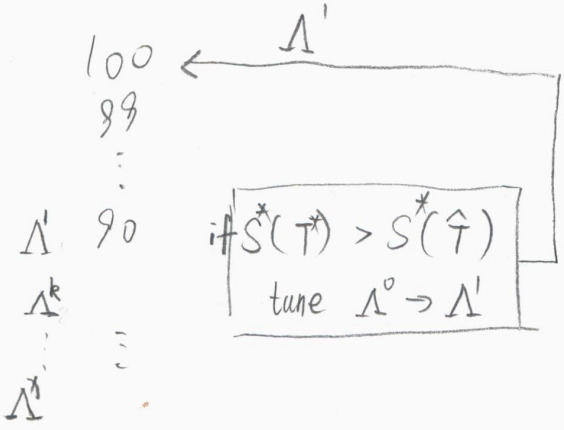
that is, choose i, j that minimizes

$$\boxed{S(t_i \cup t_j) - S(t_i) - S(t_j)}$$

And merge t_i and t_j

Problem in our Algorithm

For example, 100 clusters at the begining, use Λ^0



 Aim

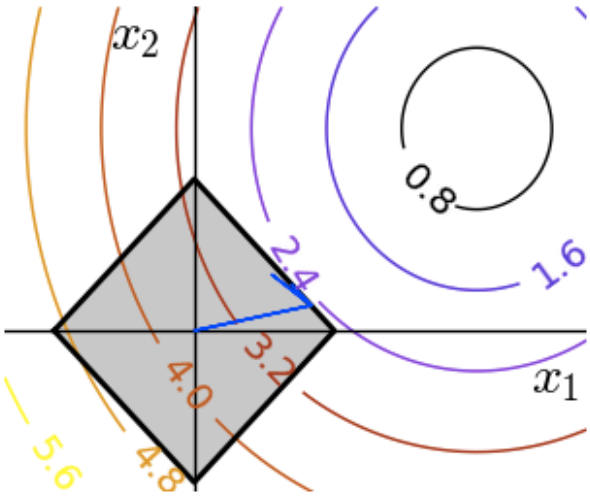
Ranking MIRA We use a variant of MIRA (Margin Infused Relaxed Algorithm), a relaxed, online maximum margin training algorithm (Crammer & Singer 2003). We updates the parameter vector with three constraints: (1) the better neighbor must have a higher score by a given margin, (2) the change to Λ should be minimal, and (3) the inferior neighbor must have a score below a user-defined threshold τ (0.5 in our experiments). The second constraint is to reduce fluctuations in Λ. This optimization is solved through the following quadratic program:

$$\Lambda^{t+1} = \underset{\Lambda}{\operatorname{argmin}} ||\Lambda^t - \Lambda||^2 \text{ s.t.}$$

$$S(N^*(T), \Lambda) - S(\hat{N}(T), \Lambda) \geq 1$$

$$S(\hat{N}, \Lambda) < \tau$$

```
Optimization terminated successfully.    (Exit mode 0)
            Current function value: 2.47487373504
            Iterations: 5
            Function evaluations: 20
            Gradient evaluations: 5
```



Package Scipy in Python