돌아가는 방식

돌아가는 작동에는 크게 3 가지의 경우가 있습니다.
1. db 에 데이터가 충분치 않을 경우
→ 무조건 성공, 로그인 데이터를 db 에 추가
2. db 에 데이터의 크기가 최대치에 도달한 경우
→ 로그인 성공 시 db 의 데이터와 비교한 후 db 의 가장 오래된 데이터를 삭제, 입력된 로그인 데이터를 삽입
3. db 의 데이터의 개수는 충분하지만 최대 개수는 아닌경우
→ db 에 로그인 데이터가 삽입

3 의 기준으로 프로그램의 작동 과정을 알아보면 다음과 같다.
생성자를 통해 옵션을 부여합니다. 옵션으로는 작성한 프로그램을 사용할지 GMM 을 사용할지, 또는 k-means 옵션이나 디버깅 모드 등이 있습니다.

그후 login 함수를 통해 사용자로 부터 아이디와 비밀번호를 입력받습니다.
입력받은 아이디와 비밀번호가 실제 아이디 비밀번호와 일치하지 않는다면 일치할때까지 받습니다.
키보드의 입력 시간을 저장한 후 3 가지의 특징값을 추출합니다. 키들을 누른 간격, 한 개의 키를 누르고 땔때까지의 시간, 전체 입력시간이 아이디와 비밀번호로에서 각각 추출하여 6 개의 특징값들을 한 개의 어레이에 저장합니다. 이때 키들이 누른 간격에는 log 를 씌웁니다. 이는 다른 간격에 비해 키들의 눌린 간격은 평균적으로 짧지만 약간의 실수로 몇배의 길이가 될 수 있다는 가정 하에 사용한 처리이며 실제로도 log 을 씌운 것이 더 높은 정답률을 보였다.

db 에 저장된 데이터들을 불러옵니다. 이 데이터 중 30%은 검증을 하는 데이터로 사용하고 70%은 검증을 당하는 데이터로 사용됩니다.

그 다음 Kmeans 옵션을 사용했을 경우 DB 의 70%의 데이터들을 k 개로 평균을 냅니다. 이는 비교할 데이터의 개수가 많아지면서 예외적인 데이터가 침입자의 데이터와 우연히 일치할 확률이 높아지는 것을 방지하기 위함입니다.

평균이 내어진 k 개의 데이터 또는 70%의 데이터와 db 의 30% 데이터를 통해 입력된 특징값과 _recognition 에서 비교를 하게 됩니다.
만약 GMM 옵션을 사용할 경우 위의 데이터를 sklearn 패키지의 GMM 을 사용하여 일치 여부를 판단합니다. 그렇지 않으면 다음과 같은 방식으로 작동합니다.

검증을 당할 데이터들의 모든 특징값들이 균일하게 적용되는 것보다는 적절한 가중치를 주는 것이 좋다고 생각했기 때문에 각 특징값 별로 표준편차를 구해 이의 역수를 가중치로 주었습니다. 이는 사용자가 습관화되지 않는 타이핑에 대해서는 검증하지 않겠다는 의미입니다. 이때 키보드에서 키들이 연속적으로 있을 경우 대부분의 사람들이 비슷한 속도로 패턴화가 되어있다는 사실을 알게되어 키보드 상에서 연속된 키가 입력될 경우 이 특징값에 대한 가중치를 0 으로 주었습니다. 검증을 할 각각의 데이터들을 검증당할 데이터들간의 차이의 제곱합을 중 최소값과 생성자에서 옵션으로 받은 threshold 를 곱한 값을 임계치로 사용합니다. 그리고 사용자가 입력한 특징값을 검증당할 데이터와 같은 방식으로 비교하여 차이값을 얻습니다. 이 차이값이 threshold 보다 작을 경우 로그인 성공으로 판단하여 보다 높을 경우 실패로 판단합니다.