1. 서론
한글 문자집에서의 어려움은 인식대상 클래스가 많고 유사문자가 많은 반면, 여러 품목의 글자를 하나의 클래스로 할 경우 그 문자의 분산이 더욱 커지게 되는 점이다. 위와 같은 문제점을 극복하기 위하여 제 2단계로 문자의 분류는 Bayes 분류기에 의한 선형한수를 적용하여 인식대상글자와 가까운 유사한 글자를 후보로 선택하고, 2단계로서 최적선형변환(Optimum Linear Transform: OLT)을 적용하여 유사한 문자들 중에서 식별을 최적화하는 새로운 특정함을 추출하여 상세분류를 행한다. 이와 같이 Bayes기의 이용하여 후보문자를 대분류하는 1단계(대분류)과정과, 최적선형변환을 적용하여 유사문자를 식별하는 2단계(상세분류)과정으로 나누어 적용함으로써 한글과 같은 대규모 문자세트에 대한 인식률을 향상시켰다. 전체 인식과정을 그림 1에 나타내었다.

2. 전처리 및 특정추출
2.1 하스토그램을 이용한 선형 정규화
일반적인 정규화 알고리즘은 입력영상의 일정한 크기의 형상으로 선형 변형시키는 선형 정규화와 영상의 스탠드로 고려하는 비선형 정규화가 있다. 본 연구에서는 선형 정규화 방법을 사용하여 변형, 보완하여 적용하였다. 이 때, 입력영상이 정규화 영상의 크기보다 큰 경우, 중요한 정보를 갖는 단선상이 지배되는 경우가 있어서 오인의 원인이 되고있다. 이와 선형 정규화의 단점을 보완하기 위하여 문자영상의 하스토그램의 변화량을 이용하였다[4].
본 논문에서는 1개의 36차원 특성을 사용하였다. 이때, 계산 과정에 따라 참�数의 본질적인 요소로 3개의 특성을 사용하였음에 관한, 노력이, 학습에 의한 확률된 구간을 이기 때문에 36 차원의 특성을 사용하였다.

3. Bayes Classifier에 의한 대분류
3.1 Bayes Classifier
1단계 인식과정에서 입력변수와 표준분산값의 계산은 Bayes 분류기에 의한 식별함수를 사용하였다. k 개의 클래스의 m차원 특성벡터에 표준화분포형태에 존재한다고 가정하고 입력변수를 x라고 하면 x가 클래스에 속할 확률은 다음과 같다.

\[ P(x) = \text{Pr}(x \mid w_k), \quad k = 1, 2, \ldots, N_c \] (1)

이것은 Bayes의 rule에 의하여

\[ P(x) = \text{Pr}(x \mid w_k) \cdot \text{Pr}(w_k), \quad k = 1, 2, \ldots, N_c \] (2)

이 되고, \( \mu_k \)는 k 클래스의 평균 벡터, \( \Sigma_k \)는 클래스의 공분산행렬이라고 하면

\[ g(x) = -\frac{1}{2} \ln |\Sigma_k| + \frac{1}{2}(x - \mu_k)^T \Sigma_k^{-1}(x - \mu_k) \quad k = 1, 2, \ldots, N_c \] (3)

여기에 -2 ln을 떼고 -m ln 2\( \pi \)을 더하고 각 폐막의 사정확률을 모두 합계하면

\[ g(x) = \frac{1}{2} (x - \mu_k)^T \Sigma_k^{-1}(x - \mu_k) + \ln |\Sigma_k| \quad k = 1, 2, \ldots, N_c \] (4)

와 같이 결정된 식을 식별함수로 사용한다.

3.2 Bayes Classifier에 의한 인식
대분류 단계에서는 받은 분포를 갖는 다중포트의 분포를 인식하기 위해서는 Bayes 기준을 사용한다. 이 경우를 통하여 다양한 분포가 어떻게 인식되는지 분석하면 각각의 계산을 할 수 있다. 이 각각의 인식기준에서는 식(4)에 나타난 것처럼 클래스의 평균과 공분산행렬을 필요로 한다. 입력은 분류된 특징 벡터로 표준복잡성이 계산된 공분산행렬을 이용하여 식(4)에 의하여 각각의 계산을 하게 된다. 이 각각의 계산은 해야 할 문제의 확률이 크고 낮아져서, 각 데이터에 대한 각 클래스의 확률이 각각의 각 클래스의 확률을 할 수 있다.

4. 최적선형반환(Bayes)에 의한 사례 분배
4.1 최적선형반환(Linear Transform: OLT)
클래스의 분할에 따르면, 클래스 내의 분산, 클래스 간의 분산행렬은 클래스의 식별능력의 적도로 사용된다(3). 클래스 내의 분산행렬(within-class scatter matrix)은

\[ S_w = \sum_{i=1}^{N} P_i E(X_i - M_i)(X_i - M_i)^T \] (5)

로 정의된다. 모든 클래스의 표준본에 대한 전체 평균벡터를 \( M_0 \)과, 클래스간의 분산행렬(between-class scatter matrix)은

\[ S_b = \sum_{i=1}^{N} P_i (M_i - M_0)(M_i - M_0)^T \] (6)

로 정의된다. 클래스의 식별력을 높이기 위해서는 클래스 내의 분산이 작다.

\[ 0 \leq P_i \leq 1 \]

\[ \sum_{i=1}^{N} P_i = 1 \]

\[ P_i \geq 0 \]

\[ M_0 = E[X] = \sum_{i=1}^{N} P_i M_i \]

로 정의된다.

\[ M_i = E[X] = \sum_{i=1}^{N} P_i M_i \]

로 정의된다. 클래스의 식별력을 높이기 위해서는 클래스 내의 분산이 작다.

\[ 0 \leq P_i \leq 1 \]

\[ \sum_{i=1}^{N} P_i = 1 \]

\[ P_i \geq 0 \]

\[ M_0 = E[X] = \sum_{i=1}^{N} P_i M_i \]

로 정의된다.

\[ M_i = E[X] = \sum_{i=1}^{N} P_i M_i \]

로 정의된다. 클래스의 식별력을 높이기 위해서는 클래스 내의 분산이 작다.

\[ 0 \leq P_i \leq 1 \]

\[ \sum_{i=1}^{N} P_i = 1 \]

\[ P_i \geq 0 \]

\[ M_0 = E[X] = \sum_{i=1}^{N} P_i M_i \]

로 정의된다.

\[ M_i = E[X] = \sum_{i=1}^{N} P_i M_i \]

로 정의된다. 클래스의 식별력을 높이기 위해서는 클래스 내의 분산이 작다.

\[ 0 \leq P_i \leq 1 \]

\[ \sum_{i=1}^{N} P_i = 1 \]

\[ P_i \geq 0 \]

\[ M_0 = E[X] = \sum_{i=1}^{N} P_i M_i \]

로 정의된다.

\[ M_i = E[X] = \sum_{i=1}^{N} P_i M_i \]

로 정의된다. 클래스의 식별력을 높이기 위해서는 클래스 내의 분산이 작다.

\[ 0 \leq P_i \leq 1 \]

\[ \sum_{i=1}^{N} P_i = 1 \]

\[ P_i \geq 0 \]

\[ M_0 = E[X] = \sum_{i=1}^{N} P_i M_i \]

로 정의된다.

\[ M_i = E[X] = \sum_{i=1}^{N} P_i M_i \]

로 정의된다. 클래스의 식별력을 높이기 위해서는 클래스 내의 분산이 작다.

\[ 0 \leq P_i \leq 1 \]

\[ \sum_{i=1}^{N} P_i = 1 \]

\[ P_i \geq 0 \]

\[ M_0 = E[X] = \sum_{i=1}^{N} P_i M_i \]

로 정의된다.

\[ M_i = E[X] = \sum_{i=1}^{N} P_i M_i \]

로 정의된다. 클래스의 식별력을 높이기 위해서는 클래스 내의 분산이 작다.

\[ 0 \leq P_i \leq 1 \]

\[ \sum_{i=1}^{N} P_i = 1 \]

\[ P_i \geq 0 \]

\[ M_0 = E[X] = \sum_{i=1}^{N} P_i M_i \]

로 정의된다.

\[ M_i = E[X] = \sum_{i=1}^{N} P_i M_i \]

로 정의된다. 클래스의 식별력을 높이기 위해서는 클래스 내의 분산이 작다.

\[ 0 \leq P_i \leq 1 \]

\[ \sum_{i=1}^{N} P_i = 1 \]

\[ P_i \geq 0 \]

\[ M_0 = E[X] = \sum_{i=1}^{N} P_i M_i \]

로 정의된다.

\[ M_i = E[X] = \sum_{i=1}^{N} P_i M_i \]

로 정의된다. 클래스의 식별력을 높이기 위해서는 클래스 내의 분산이 작다.

\[ 0 \leq P_i \leq 1 \]

\[ \sum_{i=1}^{N} P_i = 1 \]

\[ P_i \geq 0 \]

\[ M_0 = E[X] = \sum_{i=1}^{N} P_i M_i \]

로 정의된다.

\[ M_i = E[X] = \sum_{i=1}^{N} P_i M_i \]

로 정의된다. 클래스의 식별력을 높이기 위해서는 클래스 내의 분산이 작다.
5.2 실험결과

5.2.1 유사문자세트의 크기에 따른 인식률 비교

1단계 인식 후 1위 후보자의 유사세트를 선택할 때 세트가 포함하는 문자수가(N)에 따라 인식률을 비교하였다. 그림 2는 유사문자 후보들의 빈도순으로 5개에서 30개까지 유사문자 세트 크기에 따른 인식률을 나타내고 있다. 유사세트의 문자수가 20개 이상에서는 안정된 인식률을 보이고 있다. 15개 이하의 유사문자 세트는 글로벌 감소에 따른 특성량 수의 감소로 인해 낮은 인식률을 나타내고 있다. OLT의 최대 특성량 수는 1문자 2.5개이다. 최적선형변환을 하여 20개에서 30개의 차원을 사용하여 인식하더라도 현재의 36차원을 모두 사용하는 1단계의 인식률보다 높은 인식률을 나타내는 것을 확인할 수 있다. N=20일 때 최고 인식률(92.9%)을 얻었으며 이것은 Bayes 거리만에 의한 인식률(91.06%)보다 1.8% 개선되었다.

그림 2. 유사문자세트의 크기에 따른 인식률 비교

6. 결론 및 방향

한글 문자인식의 경우 인식방식과 글자체가 많은 영향을 미친다. 본문을 고려하여 인식방법이나 최적선형변환을 사용할 때, 제1단계에서는 각 글자체의 분포를 고려한 Bayes 거리에 의한 인식방법이 대체로 유용하다. 제2단계로, 고유한 유사문자 세트 내에서는 유사문자들의 변형을 충분히 고려한 최적선형변환을 이용하는 것이 더욱 효과적이다. 이와 같이 단계별로 적절한 인식방법을 적용함으로써 대규모 문자세트의 효율적인 인식이 가능하다.

발전방향으로서 유사문자세트의 선정과정에서 고정적인 유사문자수의 크기를 문자들의 특성에 따라 가변적으로 조정할 필요가 있다. 또한 1단계 인식 후 5위 후보 또는 인식률이 99%이상인 것을 고려하여 유사문자세트를 가정해 보고 싶어 5단계에서 후보 문자들로 구성하면 유사문자에 입력문자와 존재하지 않아도 오인되지 않는 것을 포함할 수 있으나 생각된다. 또한 단계별의 인식과정에서 갱신방향이 많아지기 때문에 인식속도를 향상할 수 있도록 알고리즘의 고속화 연구가 필요하다.

참고문헌