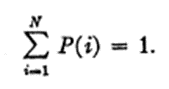
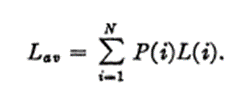
Bir sinyali bir mesajı iletmek 1952’lerde olduğu gibi günümüzde de çok önemlidir. Bir mesajı iletirken hem alıcının hemde vericinin bu işi en az hata ile en hızlı şekilde yapması gerekmektedir. Bu araştırmamda 1950’lerde ortaya atılmış bir metodu inceledim. Ve bu problemi iletmek için DAVID A. HUFFMAN bir metod önermiştir. Huffman, mesajların iletimi için mesajın kendisini iletmek yerine, mesajı anlatacak sembol dizilerini iletmenin daha güvenilir, hızlı ve kolay olabileceği fikrini ortaya atmıştır. Elimizdeki mesajı iletecek sembol türlerinden çok sayıda varsa, bazı mesajların birden fazla sembol kullanması gerekebilir. Her sembolün iletilmesi için aynı süre geçtiğini varsayarsak, bir mesajın iletim süresi o mesajın sembol uzunluğu ile doğru orantılıdır. Huffmana göre verilen bir mesajla ilişkilendirilen sembol veya semboller dizisine "mesaj kodu" adı verilecektir. İletilebilecek tüm mesaj sayısı "mesaj topluluğu" olarak adlandırılacaktır. Topluluğun her mesajı için kodun anlamı hakkında verici ve alıcı arasındaki karşılıklı anlaşmaya "topluluk kodu" adı verilir. Her mesajı için kodun anlamı hakkında verici ve alıcı arasındaki karşılıklı anlaşmaya "topluluk kodu" adı verilir.

Mesaj sayısı N olarak adlandırılır. P(i) i. Mesajın olasılığı olur. Ve bütün bu olasıkların toplamı 1 sonucunu verir.



Bir mesajın uzunluğu, L(i), kendisine atanan kodlama basamaklarının sayısıdır. Bu nedenle, ortalama mesaj uzunluğu



Formülü ile ifade edilir.

Bu metod ile, sonlu sayıda N’den ve belirli sayıda kodlama basamağından oluşan bir mesaj topluluğu için, mümkün olan en düşük ortalama mesaj uzunluğunu veren bir topluluk kodu olarak tanımlanacaktır.

Bir mesaj kodunda aşağıdaki temel kısıtlamalar uygulanmalıdır.

* Hiçbir iki mesaj aynı kodlama rakamlarından oluşmayacaktır.
* Mesaj kodları, bir mesaj dizisinin başlangıç noktası bilindiğinde, bir mesaj kodunun nerede başlayıp nerede bittiğini belirtmek için ek bir gösterge gerekmeyecek
* hiçbir mesajın, kodu, daha büyük uzunluktaki herhangi bir mesaj kodunun ilk parçası olarak belirlenmeyecektir.

Bunu açıklamak gerekirse;

A-B-C-D Karakterlerinden oluşan bir mesajın iletilmesi için öncelikle gönderilecek mesajın içinde bu karakterlerin kullanılma sıklığı yani frekansı hesaplanmalı. Bunu yazının devamında derinlemesine işleyeceğiz. Şimdi açıklamaya geri dönelim. Frekans hesaplandıktan sonra A B C D karakterleri için bir sembol atanmalı. Varsayımsal olarak rastgele bir sembol belirleyeceğiz.

A = 01

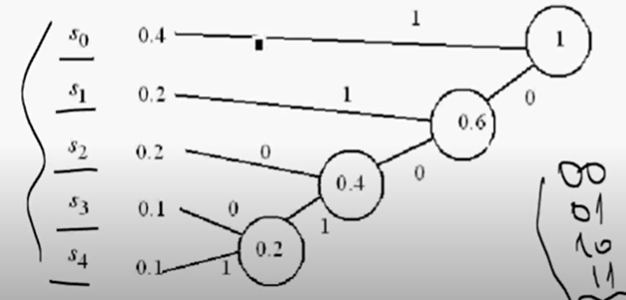
B = 102

C = 111

D = 202

Olarak belirleyelim. Mesajımızın “ABACCD” olsun. Bunu sonucunda 0110201111111202 iletilecek mesaj olacaktır. Bu mesajın için alıcının bilmesi gereken tek şey topluluk kodudur. Alıcı bu topluluk kodunu kullanarak mesajı 01-102-01-111-111-202 kısımlara bölerek geriye doğru gidebilir. Bunun sonucunda ise “ABACCD” mesajına tekrar ulaşır.

Bu makalede en önemli şeylerden biri optimum bir kod için, verilen bir mesaj kodunun uzunluğu asla daha olası bir mesaj kodunun uzunluğundan daha az olamaz. L(N) uzunluğu ile kodlanmış iki mesajın hiçbirinin L(N) -1 sıralı aynı başlangıç sinyaline sahip olmadığı bir optimum kod varsayalım. Optimum bir kod için, L(N) uzunluğundaki bu mesajlardan hiçbiri, diğer kodlara karşılık gelen herhangi bir kodun başlangıcına sahip olamaz. Bu durumda tüm mesaj grubunun son rakamı önemli hale gelir.



Anlaşılması daha kolay olsun diye bu tekniği bir örnekle açıklamaya çalışacağım. İhtimali (frekansı) bilinin mesajın en yüksek ihtimalden en düşük ihtimale sıralıyoruz. An alt iki olasılık toplanıp bir balon yapılır ve shannon a göre üst tarafa 0 alt tarafa 1 yazılır. Daha sonra diğer olasıklıla (s2) ile baloncuk toplanır. Üst tarafa 0 alta ise 1 yazılır. Tekrardan bir diğer ihtimal (s1) ile baloncuk toplanır ve sonra ihtimali yüksek olana, yani üst tarafa, 0 ihtimali düşük olana, yani alt tarafa 1 yazılır. Daha sonra diğer ihtimal ile (s0) baloncuk toplanırsa 1 elde edilir.

Artık iletilecek mesajdaki karakterlerin sembollerini atayabiliriz. Bunun için 1 den yola çıkıp karaktere doğru gidilmelidir. Örneğin S3 karakteri için 0-0-1-0 yolu izlenir.

A 🡪S0 = 1

B 🡪 S1 = 01

C 🡪 S2 = 000

D 🡪 S3 = 0010

E 🡪 S4 = 0011

Sonucuna ulaşılır. Ve yukarda görüldüğü üzere bütün kısıtlamaları da sağlamaktadır. Mesela Hiçbir sembol başka bir sembolün başlangıcını temsil etmemektedir.

Şimdi örneğimizin Lav değerini hesaplayabiliriz. Bunun için ihtimallerle karakter sayısını çarpıp toplayacağız. 0.4x1 + 0.2x2 + 0.2x3 + 0.1x4 + 0.1x4 yaklaşık olarak 2.2 değeri gelir. 5 karakterli bir mesajı normal şartlarda 3 bit ile karşı tarafa iletebilirken şuan bu bit değerini 2.2 ye düşürmüş olduk.

Şimdi bu semboller ile iletilmiş mesajı ve standart mesajı karşılaştıralım. Standart semboller için 3 bite ihtiyacımız var. Karşılıkları;

A 🡪 S0 = 001

B 🡪 S1 = 010

C 🡪 S2 = 011

D 🡪 S3 = 100

E 🡪 S4 = 101 olsun

Mesajımız “ABAABCBACDE” olsun

Standart : 001010001001010011010001011100101

Method : 101110100001100000100011

Makalenin sonucu ise verilen örnekte görülüğü gibi mesajımız kısalmış oldu ve bu sayede hatalar azalır ve mesajın iletimi daha kolay olur. Bu teknik günümüzde bir çok alanda kullanılmaktadır. Bunlardan en önemli alan veri sıkıştırma alanıdır. Ayrıca png, jpeg, mp3 gibi dosya formatlarında, ayrıca HTTP'nin standard sıkıştırma metotlarında kullanılmıştır.