Yaramaz E-Postaların Süzülmesinde, Karar Destek Makineleri, Naïve Bayes ve Bellek Tabanlı Öğrenme Yöntemlerinin Karşılaştırılması

G. Eryiğit gulsen@cs.itu.edu.tr
C. Tantuğ cuneyd@cs.itu.edu.tr
E. Adalı adali@cs.itu.edu.tr

İstanbul Teknik Üniversitesi
Bilgisayar Mühendisliği Bölümü

Özetçe

Bu makalenin amacı, yaramaz (spam) epostaları, normal e-postalardan ayırma süreci için, karar destek makineleri (Support Vector Machines - SVM), bellek tabanlı öğrenme (Memory Based Learning - MBL) ve Naïve Bayes (NB) yöntemlerinin karşılaştırmalı değerlendirmesini yapmaktır. Yaramaz e-postaların süzülmesinde kullanılan yöntemleri karşılaştıran birçok çalışma olmasına karşın, bu çalışmaların büyük çoğunluğu, farklı veri kümeleri kullandıklarından karşılaştırılabilir nitelikte değildir. Bu çalışmada, SVM, MBL ve NB yöntemleri karşılaştırılırken, herkesin erişimine açık olan ortak bir derlem (corpus) olan LINGSPAM derlemi kullanılmıştır. MBL ve NB yöntemleri, önceki çalışmalarda bu veri kümesi üzerinde sınandığı için, önceki deneylerden elde edilen en iyi parametreler ufak değişikliklerle kullanılmıştır. Ancak SVM yönteminin en iyi sonucu vermesini sağlamak için çok sayıda deney yapılmıştır. Çalışmamızda bir e-postanın, yaramaz olarak tanınması durumunda, bu e-postaya nasıl davranılacağına ilişkin senaryo önerileri verilmiş ve gerçeklenen sınıflandırıcıların hatalı çalışması durumunda ilgili senaryolara göre ortaya çıkabilecek hataların bedeli göz önüne alınarak bu üç sınıflandırma yöntemi değerlendirilmiştir. Ortaya çıkan sonuçlarda, SVM yönteminin hata bedelinin sıfır olduğu ya da yüksek olduğu senaryolar için başarımının diğer yöntemlerden daha iyi olduğu görülmüştür. Ancak hata bedelinin çok yüksek olması durumunda ise NB yöntemi en iyi sonucu vermiştir.

Abstract

This paper presents a comparison of support vector machines (SVM), memory-based learning (MBL) and Naïve Bayes (NB) techniques for the classification of legitimate and spam mails. Although there are a number of method-comparative studies regarding spam mail filtering, most of the studies are tested on separate data sets. In order to evaluate the effectiveness of SVM, MBL and NB methods, we have used a common publicly available corpus (LINGSPAM). As MBL and NB methods are previously tested with this corpus, the obtained best parameters are used in the experiments with few changes. On the other hand, intense experiments are made to find the best attribute dimensions with SVMs. Results show that SVM has significantly better performance for no-cost and high-cost cases, but NB performs best when the cost is extremely high.

1. Giriş

Geçtiğimiz son 10-15 yıl boyunca internet ve e-posta kullanıcı sayısının hızla artması, pazarlamacıların e-postaları etkin bir pazarlama ve reklam aracı olarak kullanmalarına olanak sağlamıştır. E-posta göndermenin çok kolay, daha önemlisi ucuz olmasının sonucu, istenmeyen e-postalar kullanıcıların posta kutularını doldurmaya, internet iletişimi için kullanılan hatların boşa harcanmasına ve sunucuların yükünün artmasına neden olmaktadır. Yaramaz (spam) e-postalar adı verilen bu istenmeyen mektupların tanınması ve

süzülmesi ile ilgili çalışmalar henüz çok yeni sayılabilir. Teknik anlamda, yaramaz ve normal e-postaların otomatik olarak sınıflandırılması ile ilgili ilk çalışma 1998 yılında Sahami ve ark. tarafından yapılmıştır [1].

Aslında, yaramaz e-postaların tanınması konusu, iki sınıflı bir sınıflandırma sorunudur. Bu yüzden makine öğrenmesi tekniklerinin bu amaçla kullanılmaları uygun düşmektedir. Şimdiye dek yaramaz mektupların tanınmasında kullanılan sınıflandırma yöntemleri Naïve Bayes (NB) [1,2], bellek tabanlı öğrenme yöntemleri (MBL) [3], destek (boosting) ağaçları [4] ve karar destek makineleridir (SVM) [5]. Bu yöntemlerin tamamı, sınıflamaları yapılmış eğitim derlemleri kullanılarak, sınıflandırmanın nasıl yapılacağına ilişkin bilgileri öğrenirler. Daha sonra gelen örnekleri (yeni gelen e-postayı), öğrenmiş oldukları bu bilgilere göre sınıflandırırlar. Bir başka deyişle e-postayı yaramaz ya da normal sınıf kutusuna atarlar. Bu yönteme, gözetimli öğrenme adı verilmektedir.

Sınıflandırma sürecinde her e-posta bir örnek olarak değerlendirilir ve her bir örnek için bir nitelik vektörü oluşturulur. Bu vektördeki her bir nitelik bir sözcüğü temsil etmektedir. Bir niteliğin değeri olarak, o niteliğin temsil ettiği sözcüğün, o örnekte (e-postada) geçip geçmediği gibi ikili bir bilgi tutulabileceği gibi o örnek içerisinde kaç kez geçtiği gibi sayısal bir bilgi de tutulabilir. Bu farklı iki yaklaşım "ikili değer" ve "sayısal değer" olarak adlandırılır [6]. Yaramaz e-postaların saptanması sürecinde kullanılan ve yukarıda söz edilen yöntemler, nitelik vektörlerini oluştururken ikili değer yöntemini kullanmışlardır. Sayısal değer temsil yönteminin kullanılabilir hale gelebilmesi için daha çok çalışmanın gerektiği belirtilmektedir [6].

Yaramaz e-postaların sınıflandırılması yöntemi her ne kadar bir metin sınıflandırma yöntemi olsa da, iki temel açıdan farklılık göstermektedir [3]. İlk olarak, yaramaz e-postaların konuları ve içerikleri "herşey" olabileceği için geniş bir kapsama sahiptir. İkinci olarak, bu sınıflandırma, hatalı sınıflandırma bedelinin farklı olduğu bir alandır. Normal bir e-postanın sınıflandırıcı tarafından yaramaz olarak algılanıp silinmesi, yaramaz bir e-postanın süzgeçten geçerek posta kutusuna girmesinden çok daha zararlıdır . Sonuç olarak, gerçeklenecek olan iki sınıflı sınıflandırıcının yapacağı hata türlerinin ($Y \Rightarrow N$, $N \Rightarrow Y$) bedelleri yansımalı (simetrik) değildir.

Bu hata bedelinin belirlenmesindeki ölçüt, tasarlanacak olan sistemin yaramaz e-postalara nasıl davranacağının belirlenmesidir. Yakaladığı yaramaz e-postaları silen bir sınıflandırıcının hata bedeli çok yüksek iken, yakaladığı yaramaz e-postaları sadece işaretleyen bir sınıflandırıcının hata bedeli daha düşük hatta sıfır bile olabilir.

Bu konuda yapılan çalışmaların [1, 2, 3, 4, 5] sonuçları birbirleri ile karşılaştırılamamaktadır. Çünkü bu çalışmalardaki yöntemler ortak bir eğitim ve sınama kümesini kullanmamıştır. Ayrıca bu çalışmaların çoğunda hata bedeli göz önüne alınmamaktadır.

Bu çalışmamızın hedefi, daha önceden farklı veri kümeleri üzerinde denenmiş yöntemleri bir arada, ortak bir veri kümesi üzerinde eğiterek denemek ve yöntemlerin başarımlarını, hata bedellerini de hesaba katarak değerlendirmektir. Bu yöntemler; Drucker ve ark. tarafından hata bedelini hesaplamadan kullanılan karar destek makineleri yöntemi (SVM) [6], Sakkis ve ark. tarafından hata bedelini hesaplayarak kullanılan Naïve Bayes (NB) yöntemi ve Bellek Tabanlı Öğrenme (MBL) yöntemidir. Anılan ikinci çalışmada, hata bedeli yüksek olduğu zaman MBL yönteminin daha iyi sonuç verdiği söylenmektedir. Bizim çalışmamızın sonucunda, yaramaz bir e-postayı normal olarak işaretleme hatasının bedeli, normal bir e-postayı yaramaz olarak işaretleme hatasının bedeli ile eşit olduğu zaman SVM'nin diğer iki yöntemden daha başarılı olduğunu gösterilmiştir. Ayrıca, bu hata bedelinin yüksek olduğu durumlarda da SVM'nin başarılı olmasına karşın, hata bedelinin çok çok yüksek olduğu durumlarda NB yönteminin, diğer iki yöntemden daha iyi sonuçlar verdiği gösterilmiştir.

Makale içinde, bu çalışmada kullanılan derlem tanıtılmış; niteliklerin seçilme ve örneklerin temsilinin nasıl yapıldığına ilişkin bilgiler verilmiştir. Daha sonra, kullanılan sınıflandırma yöntemleri kısaca tanıtılmış; ardından her yöntemin sonuçları verilmiştir. Son bölümde ise

yapılan çalışmaların yorumlarına ve gelecek çalışmalar hakkında bilgilere yer verilmiştir.

2. Derlem

Yapılan deneyler, dil bilimi ile ilgili bir e-posta listesine gelen normal ve yaramaz mektuplardan oluşan, herkese açık [1] "Ling-Spam" isimli bir derlem kullanılarak yapılmıştır [2]. Derlemin içindeki metinlerin dili İngilizcedir. Bu derlemin dört sürümü bulunmaktadır. Bazı sürümlerde metinlerdeki her sözcüğü, eklerinden arındıran (sadece kökünü bırakan) bir "kök bulucu" kullanılarak sözcük uzayının boyutu düşürülmüştür. Benzer şekilde çok fazla ayırıcı bilgi taşımayan ancak sıkça kullanılan 100 sözcüğü (and, a, an gibi) içeren bir yasak listesi hazırlanmış ve bu listedeki sözcükler metinlerden çıkartılmıştır. Bu iki yardımcı aracın beraber kullanıldığı ve kullanılmadığı toplam dört farklı sürüm bulunmaktadır. Bizim çalışmamızda yeğlediğimiz sürüm, "kök bulucu"nun ve "yasak listesi"nin kullanıldığı sürümdür. Bu sürümün, diğerlerine göre daha yüksek başarılar gösterdiği belirtilmektedir [2]. Kullanılan derlem 2412'si normal, 481 tanesi yaramaz olmak üzere toplam 2893 mektuptan oluşmaktadır. Derlem, aynı oranda yaramaz ve normal mektup içeren 10 parçaya bölünmüştür. Yapılan her bir deney 10 kez tekrarlanmış; her defasında mevcut olan 10 parçadan 9 tanesi eğitim amaçlı kullanılmış; kalan 1 parça da sınama amaçlı kullanılmıştır.

3. Niteliklerin Seçilmesi ve Örneklerin Temsili

Giriş bölümünde kısaca değinildiği gibi çalışmamızda ikili değer modeli kullanılmıştır. Bu modelde, her bir e-posta, bir örnek olarak değerlendirilmiş ve her bir örnek için oluşturulan nitelik vektörü, seçilen bir dizi sözcüğün o örnek içerisinde var olup olmadığına ilişkin 0 veya 1 değerlerini içerecek biçimde oluşturulmuştur. Belirtilen bu "bir dizi" sözcük seçilirken, her bir sözcüğün taşıdığı "ortaklık bilgisi" (KB- Mutual Information) hesaplanmıştır. Bu değer bir anlamda, bu sözcüğün, derlem içerisinde ne kadar bir ayırıcılığa sahip olduğunu göstermektedir. Aşağıda, bu değerin hesaplanmasında kullanılan formül verilmiştir. En yüksek KB (MI) değerine sahip "n" adet sözcük, nitelik olarak seçilmiştir ve sadece bu "n" adet sözcüğün örneklerde olup olmadığına bakılmıştır. Her bir örnek $\vec{X}$ =<x1,x2,…..xn> şeklinde bir nitelik vektörü ile ifade edilmiştir.

$$\mathbf{MI}(X,C) = \sum_{x\in\{0,1\},c\in\{yaramaz,\ normal\}} P(X=x,C=c).\log_2 \frac{P(X=x,C=c)}{P(X=x).P(C=c)} \quad (1)$$

4. Sınıflandırma Yöntemleri

Bu bölümde, çalışmamızda kullanılan SVM, MBL ve NB yöntemleri hakkında kısa bilgiler verilecektir. Bu yöntemler ile ilgili daha fazla bilgi için şu kaynakların incelenmesi önerilir: Sahami ve ark. [1], Androutsopoulos ve ark. [2], Sakkis ve ark. [3], Drucker ve ark. [5]. Yöntemlerin ayrıntıları verilmeden önce, hata bedelinin önem kazandığı yaklaşımlarda, sınıflandırmanın nasıl değerlendirildiği konusuna değinmeyi yararlı buluyoruz. Normal bir e-postanın yanlışlıkla yaramaz olarak sınıflandırılması, yaramaz bir e-postanın normal olarak işaretlenmesinden çok daha önemli bir hata olarak kabul edilmektedir. Normal→Yaramaz hatasının, Yaramaz→Normal hatasından λ kat daha önemli olduğu söylenebilir. Bu durumda sınıflandırıcının, bir e-postayı "yaramaz" olarak sınıflandırması veya adlandırması için aşağıdaki koşulun sağlanması gereklidir:

$$\frac{P(C=yaramaz \mid \vec{X}=\vec{x})}{P(C=normal \mid \vec{X}=\vec{x})} > \lambda \quad (2)$$

E-posta sınıflandırması iki sınıflı bir sınıflandırma olduğu için aşağıdaki bağıntı yazılabilir.:

$$P(C=yaramaz \mid \vec{X}=\vec{x}) = 1 - P(C=normal \mid \vec{X}=\vec{x})$$

Yukarıda da gösterildiği gibi, yeni gelen bir $\vec{x}$ örneği ancak ve ancak güvenilirlik seviyesi

[1] *Ling-Spam http://www.aueb.gr/users/ion/*

$$\frac{P(C = yaramaz \mid \vec{X} = \vec{x})}{1 - P(C = yaramaz \mid \vec{X} = \vec{x})} > \lambda$$

$$P(C = yaramaz \mid \vec{X} = \vec{x}) > t$$

$$t = \frac{\lambda}{\lambda + 1} \qquad (3)$$

$$W_s(\vec{x}) > t$$

$W_S(\vec{x})$, λ 'nın bir fonksiyonu olan *t*'den daha büyük olursa "yaramaz" sınıfı olarak etiketlenebilir.

Tüm yöntemler, 10-katlı çapraz-doğrulama tekniği kullanılarak eğitilmiş ve sınanmıştır. Toplam veri kümesi 10 eş boyutlu parçaya bölünmüş, her adımda bu 10 parçadan farklı bir tanesi sınama için ayrılmış, diğer 9 parça eğitim sürecinde kullanılmıştır. Her yöntemin her denemesi için bu işlem 10 defa tekrarlanmıştır.

4.1 Naïve Bayes

Bayes ve toplam olasılık kuramından yola çıkılarak, $\vec{x}$ =<x1,x2,.....xn> nitelik vektörü olan bir örneğin (e-posta) *c* sınıfına ait olması olasılığı şöyledir:

$$P(C = c \mid \vec{X} = \vec{x}) = \frac{P(C = c).P(\vec{X} = \vec{x} \mid C = c)}{\sum_{k \in \{yaramaz,\, normal\}} P(C = k).P(\vec{X} = \vec{x} \mid C = c)} \qquad (4)$$

Uygulamada, $P(\vec{X} = \vec{x} \mid C = c)$ olasılıklarını hesaplamak, bazı kabuller yapmadan olanaksızdır. Çünkü olası $\vec{x}$ değerleri sayısı çok fazladır ve üstelik veri seyrekliği sorunu da bulunmaktadır.

$$P(C = c \mid \vec{X} = \vec{x}) = \frac{P(C = c).\prod_{i=1}^{n} P(X_i = x_i \mid C = c)}{\sum_{k \in \{yaramaz, normal\}} P(C = k).\prod_{i=1}^{n} P(X_i = x_i \mid C = c)} \qquad (5)$$

Naïve Bayes sınıflandırıcısı, belirli bir *c* sınıfı için x1, x2, ….. , xn niteliklerinin koşullu olarak bağımsız olduğu varsayımında bulunmaktadır (denklem (5)).

4.2 Bellek Tabanlı Öğrenme

MBL yöntemi [7] en yakın k-komşu (K-NN) sınıflandırıcısının bir türevidir. K-NN yönteminde, her gelen yeni e-posta, kendisine en yakın k adet e-postanın sınıflarından çoğunluğa sahip olanı hangisi ise o sınıfa atanırken MBL yönteminde, yeni gelen e-posta en yakın k uzaklık içerisinde kalan e-postaların çoğunluk sınıfına atanır. Bunun sonucu olarak, eğer en yakın k uzaklık içerisinde birden çok komşu varsa, komşu sayısı k'dan fazla olacaktır. İki örnek arasındaki mesafe hamming uzaklığı kullanılarak bulunur. İki $\vec{x}_i$ ve $\vec{x}_j$ örneği arasındaki d($\vec{x}_i, \vec{x}_j$) uzaklığı denklem (6)'daki gibi hesaplanır.

$$\vec{x}_i = \langle x_{i1}, x_{i2}.....x_{in} \rangle \quad and \quad \vec{x}_j = \langle x_{j1}, x_{j2}.....x_{jn} \rangle$$

$$\delta(x, y) \equiv \begin{cases} 0, & eğer \quad x = y \quad ise \\ 1, & diğer \quad durumlarda \end{cases} \qquad (6)$$

$$d(\vec{x}_i, \vec{x}_j) \equiv \sum_{r=1}^{n} \delta(x_{ir}, x_{jr})$$

Bir $\vec{x}$ örneğinin c sınıfına ait olmasının güvenilirlik seviyesi denklem (7)'deki gibi hesaplanır. Bu denklemde C($\vec{x}_i$), i numaralı komşunun sınıfını belirtir. Bu hesaplamadan sonra güvenilirlik seviyeleri [0-1] aralığına getirilip, denklem (3) bir e-postayı yaramaz olarak sınıflandırmak üzere kullanılabilir.

$$W_c(\vec{x}) = \sum_i \left(1 - \delta(c, C(\vec{x}_i))\right) \qquad (7)$$

MBL'in başarısı bazı ağırlaştırma yöntemleri kullanılarak arttıralabilir. WMBL (Ağırlaştırılmış MBL) için nitelik ve uzaklık ağırlaştırma yöntemleri kullanılmıştır.

4.2.1 Uzaklık Ağırlaştırma

Uzaklık ağırlaştırma denklem (8)'i uygulayarak, giriş örneğine daha yakın komşuları daha önemli sayar.

$$W_c(\vec{x}) = \sum_i f_n(d(\vec{x},\vec{x}_i)).(1-\delta(c,C(\vec{x}_i)))$$
$$f_n(d) = \frac{1}{d^3} \quad (8)$$

4.2.2 Nitelik Ağırlaştırma

MBL'de, gerçekte öyle olmamasına rağmen, bütün nitelikler eşit önemde sayılırlar. Bu nedenle WMBL'deki nitelik ağırlaştırması tüm niteliklere eşit davranmamayı hedefleyerek, denklem (9) kullanarak, her özelliğe farklı önem puanları atar. Denklem (6)'daki $d(\vec{x}_i, \vec{x}_j)$ uzaklık ölçümü denklem (9)'daki hale dönüşür.

4.3 Karar Destek Makineleri

Vapnik'in Karar Destek makineleri (SVM) [10] iki sınıf arasındaki payı (margin) en büyük yapacak ayırıcı hiperdüzlemi bulmaya çalışan, çok kullanılan etkili bir örüntü tanıma tekniğidir. SVM yüksek boyutlu veri kümeleri üzerinde çok iyi sonuçlar veren iki sınıflı bir sınıflandırma yöntemidir. SVM aşağıdaki eniyileme sorunu ile eğitilir.

$$\hat{w} = \arg\min_w \frac{1}{2}\|w\|^2 + C\sum_i \xi_i \quad (10)$$
$$y_i(d_i \cdot w + b) \geq 1 - \xi_i \qquad \xi_i \geq 0$$

Bu denklemde her d_i bir belge vektörü, her y_i +1 veya –1 değeri alan bir d_i etiketi ve w en uygun ayırıcı hiperdüzlemi tanımlayan ağırlık vektörüdür. Bu tip eniyileme sorunlarına "ilkeli" adı verilir. Eşitsizlik kısıtlarını Lagrange çarpanları yolu ile biraraya getirerek, sorunun "benzer" şekline ulaşılır.

$$\hat{w} = \arg\max_w \sum_i \alpha_i - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j (d_i \cdot d_j)$$
$$0 \leq \alpha_i \leq C \qquad \sum_i \alpha_i y_i = 0 \quad \forall i \quad (11)$$

α_i'nin optimize edilen değerleri verildiğinde en uygun hiperdüzlem söyledir:

$$\hat{w} = \sum_i \alpha_i y_i d_i \quad (12)$$

Yukarıdaki kısıtlı sorun Platt'ın ardışıl enküçük eniyileme (Sequential Minimal Optimization) [11] ve Osuna'nın yöntemi [12] gibi bazı hızlı çözme yöntemleri, quadratic programlama ile çözülebilir.

$$d(\vec{x}_i, \vec{x}_j) \equiv \sum_{r=1}^{n} w_r.\delta(x_{ir}, x_{jr})$$
$$w_r = H(C) - \sum_{x\in\{0,1\}} P(X=x).H(C|X=x) \quad (9)$$
$$H(C) = -\sum_{c\in\{yaramaz,normal\}} P(C=c).\log_2 P(C=c)$$
$$H(C|X=x) = -\sum_{c\in\{yaramaz,\,normal\}} P(C=c|X=x).\log_2 P(C=c|X=x)$$

SVM ve quadratic problem çözme yöntemleri ile ilgili ayrıntılı bilgi [8]'de bulunabilir. Çalışmamızda, bir SVM uygulaması olan LibSVM [9] kütüphanesi kullanılmıştır. LibSVM'nin en son sürümü olan LibSVM 2.6'nın iki sınıfa da dahil olma güvenilirlik seviyelerini verme özelliği vardır. Bu özellik bize SVM, MBL ve Naïve Bayes yöntemlerini farklı maliyetler için karşılaştırabilme imkânı tanır. Quadratic denklemleri çözmek için doğrusal çekirdek kullanılmıştır.

5. Sonuçlar

Bu bölümde, SVM, MBL ve NB algoritmalarının uygulanması ile elde edilen sonuçlar verilmiştir. Drucker ve ark. [5] iki sınıflı sınıflandırma yapılan durumlarda, yakalama oranı (recall rate) (15) ve tutturma oranlarının (precision rate) (16) kullanılamaz olduklarını söylemiştir. Bunların yerine yanlış uyarı oranı (false alarm rate) (13) ve kaçırma oranının (miss rate) (14) kullanılması gerektiğini vurgulamıştır. Ancak, önceki çalışmaların büyük çoğunluğu, sonuçlarını yakalama ve tutturma oranlarını kullanarak vermişlerdir. Normal bir e-postanın yaramaz olarak sınıflandırılmasının, yaramaz bir e-postanın normal olarak sınıflandırılmasından daha yüksek bedelli olduğu durumlarda, [3]'de tanımlanan bedel fonksiyonu TBO (toplam bedel oranı) (17) başarımların karşılaştırılması için uygun bir yöntem olarak görülmektedir. TBO fonksiyonunun türetimi [3]'de bulunabilir. Büyük TBO değerleri yüksek başarım anlamına gelir. Bu durum, denklem (17)'de açıkça görülebilir. TBO değeri 1'den küçük olduğunda, süzgeci kullanmamak daha iyidir. Çalışmamızda, önceki çalışmaların sonuçları ile ilişki kurabilmek için, sözü geçen üç ölçüt de verilmiştir:

$$YUO\ (Yanlış\ Uyarı\ Oranı) = \frac{N_{Y\to N}}{N_Y} \qquad (13)$$

$$KO\ (Kaçırma\ Oranı) = \frac{N_{N\to Y}}{N_N} \qquad (14)$$

$$YO\ \ (Yakalama\ Oranı) = \frac{N_{Y\to Y}}{N_{Y\to Y} + N_{Y\to N}} \qquad (15)$$

$$TO\ (Tutturma\ Oranı) = \frac{N_{Y\to Y}}{N_{Y\to Y} + N_{N\to Y}} \qquad (16)$$

$$TBO\ (Toplam\ Bedel\ Oranı) = \frac{N_Y}{\lambda N_{N\to Y} + N_{Y\to N}} \qquad (17)$$

Yöntemlerin karşılaştırılmasından önce, kullanılan derlem üzerinde en iyi sonuçları veren parametrelerin (MBL'deki k değeri ve nitelik boyutları) belirlenmesi gerekir. Androutsopoulos ve ark. [2]'de NB'nin LINGSPAM üzerinde λ=1 için nitelik boyutu dim=100, λ=9 için dim=100, λ=999 için dim=300'de en iyi sonuçları verdiğini belirtmişlerdir. Uygulamamızda, λ=999 için dim=100'ün dim=300'e göre daha iyi sonuç verdiği görülmüştür (Tablo–1).

Tablo-1: NB Nitelik Boyutu

Yöntem	Boyut	λ=999 TBO
NB	100	4.19
NB	300	0.15

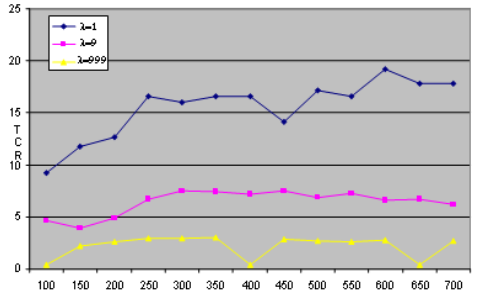
Sakkis ve ark. [3] LINGSPAM üzerinde WMBL'i kullanırken dim=600 ve k=8 komşu sayısı ile en iyi sonucu elde ettiklerini açıklamışlardır. Bizim çalışmamızda en iyi sonuçlar k=2 için elde edilmiştir (Tablo–2).

Tablo-2: WMBL "k" Değeri

Yöntem	Boyut	λ=1 TBO	λ=9 TBO	λ=999 TBO
WMBL (k=2)	600	5.87	3.37	0.15
WMBL (k=8)	600	4.86	2.00	0.38

Önceki çalışmalarda, LINGSPAM üzerinde SVM kullanılarak yapılan herhangi bir çalışma olmadığı için, denemelerimizde, en iyi TBO sonucunu veren nitelik boyutunu seçmek amacıyla 50 ile 700 arasında ellişer ellişer artan farklı sayılarda nitelik boyutları denenmiştir. Üç farklı bedel senaryosu üzerinde en iyi ortalama

TBO değerini veren Dim=600 değeri, nitelik boyutu olarak seçilmiştir. (Şekil-1)



Boy.	λ=1 TBO	λ=9 TBO	λ=999 TBO	Ort TBO
100	9.25	4.63	0.40	4.76
150	11.73	3.94	2.18	5.95
200	12.66	4.90	2.60	6.72
250	16.59	6.68	2.93	8.73
300	16.03	7.51	2.93	8.83
350	16.59	7.40	**3.04**	9.01
400	16.59	7.18	0.42	8.06
450	14.15	**7.51**	2.85	8.17
500	17.18	6.87	2.73	8.93
550	16.59	7.29	2.63	8.83
600	**19.24**	6.59	2.78	**9.54**
650	17.81	6.68	0.41	8.30
700	17.81	6.25	2.70	8.92

Şekil-1: SVM Nitelik Boyutu

NB, MBL ve SVM Karşılaştırılması

Yöntemler, hesaplanan en iyi parametreler kullanılarak sınanmış ve YUO/KO sonuçları Tablo-3'de verilmiştir. Nitelik ve uzaklık ağırlaştırmasının MBL'e katkılarının görülebilmesi için tabloya basit MBL'in sonuçları da eklenmiştir.

KO arttıkça, yanlış sınıflandırılan normal e-postaların sayısı artmakta, YUO arttıkça, yanlış sınıflandırılan yaramaz e-postaların sayısı artmaktadır. Dolayısı ile kabul edilebilir bir süzgeçte KO ve YUO'ın ikisininde olabildiğince küçük olmaları gerekir (yetkin bir süzgeçte 0 olmadırlar). Bedellerin farklı olduğu durumlarda ise KO çok daha önemlidir ve YUO'ya göre çok daha fazla cezalandırılması gerekir.

Tablo–3: Yanlış Uyarı / Kaçırma Oranları

Yöntem	Boyut	λ=1	
		YUO	KO
MBL (k=2)	600	0.397	0.0000
WMBL (k=2)	600	0.147	0.0045
NB	100	0.114	0.0029
SVM	600	0.035	0.0033
Yöntem	**Boyut**	**λ=9**	
		YUO	**KO**
MBL (k=2)	600	0.550	0.0000
WMBL (k=2)	600	0.241	0.0012
NB	100	0.160	0.0025
SVM	600	0.114	0.0008
Yöntem	**Boyut**	**λ=999**	
		YUO	**KO**
MBL (k=2)	600	0.550	0.0000
WMBL (k=2)	600	0.247	0.0012
NB	100	0.239	0.0000
SVM	600	0.360	0.0000

Bir başka değerlendirme ölçütü tutturma ve yakalama oranlarıdır. Tablo-4'de tutturma ve yakalama oranları farklı bedel değerleri ve değişik yöntemler için verilmektedir.

Tablo–4: Tutturma / Yakalama Oranları

Yöntem	Boyut	λ=1	
		YO	TO
MBL (k=2)	600	0.60291	1.00000
WMBL (k=2)	600	0.85239	0.97387
NB	100	0.88565	0.98383
SVM	600	0.96465	0.98305
Yöntem	**Boyut**	**λ=9**	
		YO	TO
MBL (k=2)	600	0.45114	1.00000
WMBL (k=2)	600	0.75883	0.99184
NB	100	0.83991	0.98536
SVM	600	0.88565	0.99532
Yöntem	**Boyut**	**λ=999**	
		YO	TO
MBL (k=2)	600	0.45114	1.00000
WMBL (k=2)	600	0.75259	0.99178
NB	100	0.76091	1.00000
SVM	600	0.64033	1.00000

λ=1 ve λ=9 bedel değerleri için SVM'nin en iyi başarımı sağladığı Tablo-4'de rahatça görülebilir. En yüksek bedel değeri λ=999 için, WMBL'in başarımı sabit kalırken SVM'nin başarımı ciddi şekilde düşmektedir.

Tablo–5: Toplam Bedel Oranı				
Yöntem	**Boyut**	**λ=1 TBO**	**λ=9 TBO**	**λ=999 TBO**
MBL (k=2)	600	2.52	1.83	1.83
WMBL (k=2)	600	5.87	3.37	0.15
NB	100	7.77	3.68	4.19
SVM	600	19.26	6.60	2.78

TBO bir yöntemin başarımını bedellerin farklı olduğu durumlarda ifade ettiği için, sonuç olarak (Tablo-5) SVM'nin, en iyi ikinci yöntem olan NB'den, bedel farkı olmadığı zaman (λ=1) neredeyse üç kat daha yüksek bir başarım sergilediği söylenebilir. Bedel değeri λ=9 durumunda da yine SVM en iyi başarımı gösteren yöntemdir. Ama yukarıdaki paragrafta da anlatıldığı gibi, bedel farkı çok yüksek olduğunda (λ=999), NB SVM'den daha başarılıdır.

6. Sonuçlar ve Gelecek Çalışmalar

Bu makale, yaramaz e-postaların süzülmesi alanında karar destek makineleri (SVM), naïve bayes ve bellek tabanlı öğrenme yöntemlerinin bedellerinin farklı olduğu durumlarda karşılaştırılmasını hedeflemektedir. Yöntemleri karşılatırabilmek ve önceki çalışmaların sonuçlarını kullanabilmek için, herkese açık olan bir e-posta listesi derlemi (LINGSPAM) kullanılmıştır. NB, MBL ve SVM yöntemleri bedellerin farklı olduğu durumlarda uygulanmıştır. Bu sırada SVM için bir kütüphane kullanılmıştır. Değerlendirme üç farklı bedel senaryosu kullanılarak yapılmıştır. SVM yönteminin bedel farkının düşük ve yüksek olduğu durumlarda farkedilir derecede daha iyi sonuç verdiği görülmüştür. Buna karşın bedel farkının aşırı yüksek olduğu durumlarda NB yöntemi en iyi başarımı göstermiştir. Sonuç olarak, çalışmamız yaramaz e-posta süzülmesi konusunda üç farklı yöntemin aynı veri kümesi üzerinde karşılaştırılmasını sağlamıştır. Çalışmamızın bir başka katkısı olarak, bedel farkının olduğu durumlarda SVM'lerin yaramaz e-postaları süzmek için uyarlanmasıdır.

Yaramaz e-postaların süzülmesi konusunda bazı örüntü tanıma teknikleri uygulanmış olsa da, tüm teknikler denenmemiştir. Yaramaz e-postaların süzülmesi konusunda diğer yöntemlerin de gerçeklenmesi ve birbirleriyle karşılaştırmaların yapılması gerekmektedir. Naïve Bayes gibi bazı basit teknikler bile beklenmedik şekilde yüksek başarım gösterebilmektedirler. Daha önceki bir çalışmada [5] lineer kernel kullanıldığı için, buradaki çalışmalarda da lineer kernel kullanılmıştır. Ancak en verimli SVM sınıflandırmasının bulunması için diğer kernel çeşitlerinin de (sigmoid, polynomial vb..) incelenmesi ve sınanması gerekmektedir.

E-postaların farklı ifade ediliş şekilleri derinlemesine incelenmemiştir. Önceki çalışmalar ile ilişki kurabilmek için ikili değer gösterim biçimi seçilmiş olmasına karşın, sayısal değer biçimi gibi bir gösterim bu yöntemlerin başarımında büyük bir etki gösterebilir. Farklı gösterimlerin etkilerini incelemek gelecekteki hedeflerimiz arasındadır. Buna ek olarak, e-posta ile ilgili ipuçları veren diğer bazı bilgiler de (domen bilgisi vb...) gösterime eklenebilir. E-postaların vektör gösterimini oluşturan seçilmiş sözcüklere ek olarak, bazı sözcük örüntülerinin varlığı da yeni nitelikler olabilir. Gelecekte olası tüm yöntemleri ve gösterimleri bedel farklılığı olan durumlarda karşılaştırmayı planlamaktayız.

Kaynaklar

[1] **Sahami, M., S. Dumais, D. Heckerman, E. Horvitz.** 1998. "*A Bayesian Approach to Filtering Junk E-Mail*". Learning for Text Categorization – Papers from the AAAI Workshop, pages 55–62, Madison Wisconsin. AAAI Technical Report WS-98-05.

[2] **Androutsopoulos I., Koutsias J., Chandrinos K.V., Paliouras G., Spyropoulos C.D.**, 2000. *"An Evaluation of Naive Bayesian Anti-Spam Filtering"*. Proceedings of the workshop on machine Learning in the New Information Age, 11th European Conference on Machine Learning (ECML 2000), Barcelona, Spain, pp. 9-17.

[3] **Sakkis G., Androutsopoulos I., Paliouras G., Karkaletsis V., Spyropoulos C.D., Stamatopoulos** P., 2003, "*A Memory-Based Approach to Anti-Spam Filtering for Mailing*

Lists", Information Retrieval 6(1), 49-73, Kluwer Publishing

[4] **Xavier Carreras, Lluís Marquez**, Boosting Trees for Anti-Spam Email Filtering (2001), Proceedings of RANLP-01, 4th International Conference on Recent Advances in Natural Language Processing

[5] **Drucker H., Wu D., Vapnik V.N.,** 1999. "*Support Vector Machines for Spam Categorization*", IEEE Transactions On Neural Networks, pages 1048-1054.

[6] **Karl-Michael Schneider,** A Comparison of Event Models for Naive Bayes Anti-Spam E-Mail Filtering, 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL 03), pp. 207-314, 2003.

[7] **Androutsopoulos I., Paliouras G., Karkaletsis V., Sakkis G., Spyropoulos C.D., Stamatopoulos P.**, 2000, "*Learning to filter Spam E-Mail: AComparison of a Naïve Bayesian and a Memory-Based Approach* ", Proc. of the workshop on Machine Learning and Textual Information Access, 4th European Conference on Principles and Practice of Knowledge Discovery in Databases, France

[8] **Duda, R.O. and P.E. Hart.** 1973. "*Bayes Decision Theory*". Chapter 2 in Pattern Classification and Scene Analysis, pages 10–43. John Wiley.

[9] **Chih-Chung Chang and Chih-Jen Lin, LIBSVM** : a library for support vector machines, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

[10] **Vapnik,** 1995. "The Nature of Statistical Learning Theory". Springer-Verlag, 1995.

[11] **J. C. Platt,** 1998. "*Sequential minimal optimization: A fast algorithm for training support vector machines*," in Advances in Kernel Method: Support Vector Learning, Scholkopf, Burges, and Smola, Eds. Cambridge, MA: MIT Press, pp. 185–208.

[12] **E. Osuna, R. Freund, F. Girosi**, 1997. "*Improved training algorithm for support vector machines*," in Proc. IEEE NNSP'97.