

Dersin içeriği

- Yapay Zeka
- Makine Öğrenmesi
- Derin Öğrenme
- Nesnelerin İnterneti
- Dijital Dönüşüm
- Artırılmış ve Sanal Gerçeklik
- **Örüntü Tanıma**
- Dijital vatandaşlık
- Robotik

Örüntü Tanıma

Örüntü: Bir nesnenin bulunduğu sınıf ya da kategoriye ait olduğunu belirten ölçülebilir veya gözlemlenebilir özellikleridir.

Bir özelliğin örüntü olarak adlandırılabilmesi için aşağıdaki şartları sağlaması beklenir.

- Sınıf içerisinde benzerlik göstermesi
- Tekrarlanabilir olması
- Gözlemlenebilir olması

Örüntü Tanıma

- Fiziksel nesne ya da olayı, önceden belirlenmiş kategorilere atama (Duda& Hart).
- Yüksek boyutlu uzayda yoğunluk fonksiyonlarını kestirme ve uzayı kategori ya da sınıf bölgelerine bölme (Fukunaga).
- Karmaşık işaretlere verilen belli örnekler ve bunlara ait doğru kararları kullanarak, gelecek yeni örnek dizileri için otomatik karar verme (Ripley).
- Ölçümlerin sınıflandırılması ya da tanımlanması ile ilgilenen bilim (Schalkoff).
- X gözlemlerine ω (sınıf etiketi) ismi verme süreci (Schürmann)

Nesnelerden elde edilen ölçüm sonuçlarından yararlanarak, nesnenin doğru sınıfa otomatik olarak atanması işlemidir.

Nesne sınıflarının özellik örüntülerinin düzenliliği otomatik olarak elde edilmektedir.

.

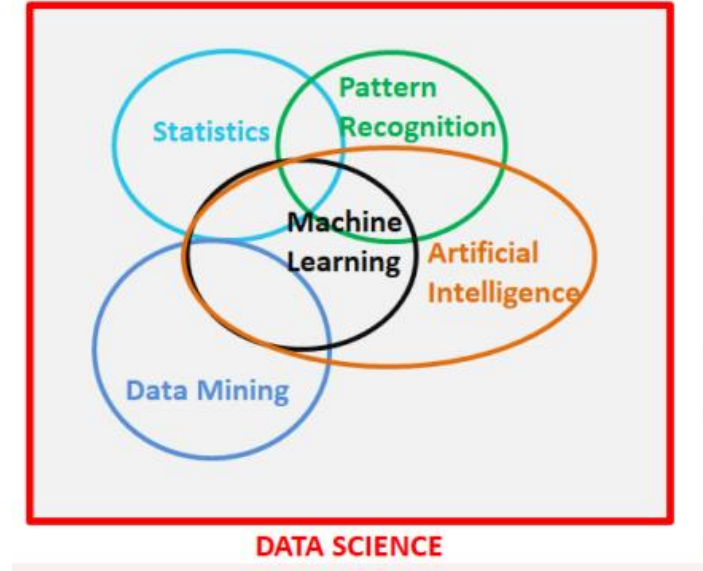
Örüntü Tanıma

Örüntü Tanıma sürecinin merkezinde olayların;

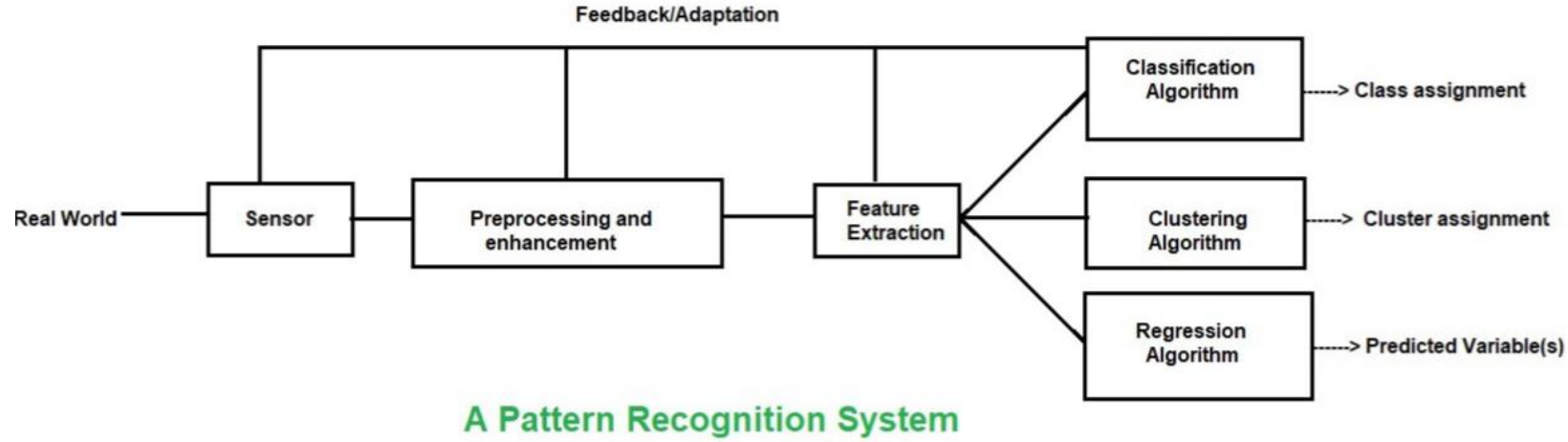
- istatistiksel bilgilere,
- geçmiş verilere veya
- makinenin belleğine

dayalı olarak sınıflandırılması yatar.

Örüntü tanıma, öncelikle alınan ölçümlerin tanımlanması ve sınıflandırılması ile fiziksel veya zihinsel süreçlerden alınan ölçümlerin tanımlanması ve sınıflandırılması ile ilgilidir.



Örüntü Tanıma



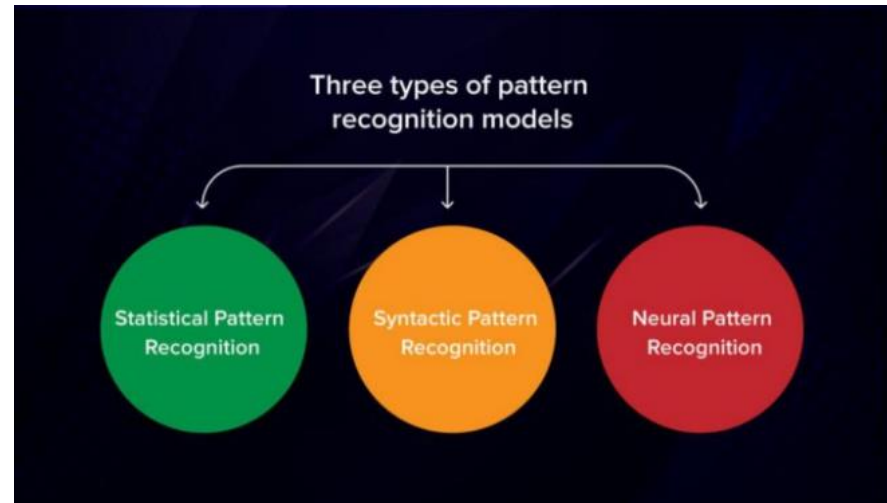
Örüntü tanımanın sonucu ya sınıf ataması ya da küme ataması ya da tahmin edilen değişkenler olabilir.

Bu nedenle, “örüntü tanıma ve sınıflandırma arasındaki fark nedir” diye sormanın bir anlamı yoktur — sınıflandırma algoritması, hedef değerin sonlu bir sınıflar kümesi olduğu denetlenen makine öğrenimi problemlerinin bir parçasıdır.

Örüntü tanıma ve bilgisayarla görme birbirinden farklı kavramlardır. Bu iki teknoloji benzer görünmekle birlikte, bilgisayarla görme teknolojisi çoğunlukla nesne algılama, görsel tabanlı öğrenme ve bölümlleme gibi görüntüleri ve görsel bilgileri işleme ve analiz etmeye odaklanır. Öte yandan örüntü tanıma, her türlü verideki örüntülerin otomatik olarak keşfedilmesini amaçlamaktadır (görsel ya da değil)

Örüntü Tanıma

- Üç tür örüntü tanıma modeli vardır:
- İstatistiksel Örüntü Tanıma: örneklerden öğrendiğinde istatistiksel tarihsel verileri ifade eder: gözlemleri toplar, işler ve bu kuralları genellemeyi ve yeni gözlemlere uygulamayı öğrenir.
- Yapısal Örüntü Tanıma: Aynı zamanda yapısal örüntü tanıma olarak da adlandırılır, çünkü ilkel olarak adlandırılan daha basit alt modellere (örneğin, sözcükler) dayanır. Kalıp, ilkeller arasındaki bağlantılar açısından tanımlanır, örneğin, sözcükler cümleleri ve metinleri oluşturur.
- Sinir Örüntü Tanıma: Sinir örüntü tanımada yapay sinir ağları kullanılır. Karmaşık doğrusal olmayan girdi-çıkı ilişkilerini öğrenebilir ve kendilerini verilere uyarlayabilirler.



Örüntü Tanıma

- **Pattern Recognition:** “the act of taking raw data and taking an action based on the category of the pattern.”
- **Common Applications:** speech recognition, fingerprint identification (biometrics), DNA sequence identification
- **Related Terminology:**
 - **Machine Learning:** The ability of a machine to improve its performance based on previous results.
 - **Machine Understanding:** acting on the intentions of the user generating the data.
- **Related Fields:** artificial intelligence, signal processing and discipline-specific research (e.g., target recognition, speech recognition, natural language processing).

Örüntü Tanıma

Recognition or Understanding?

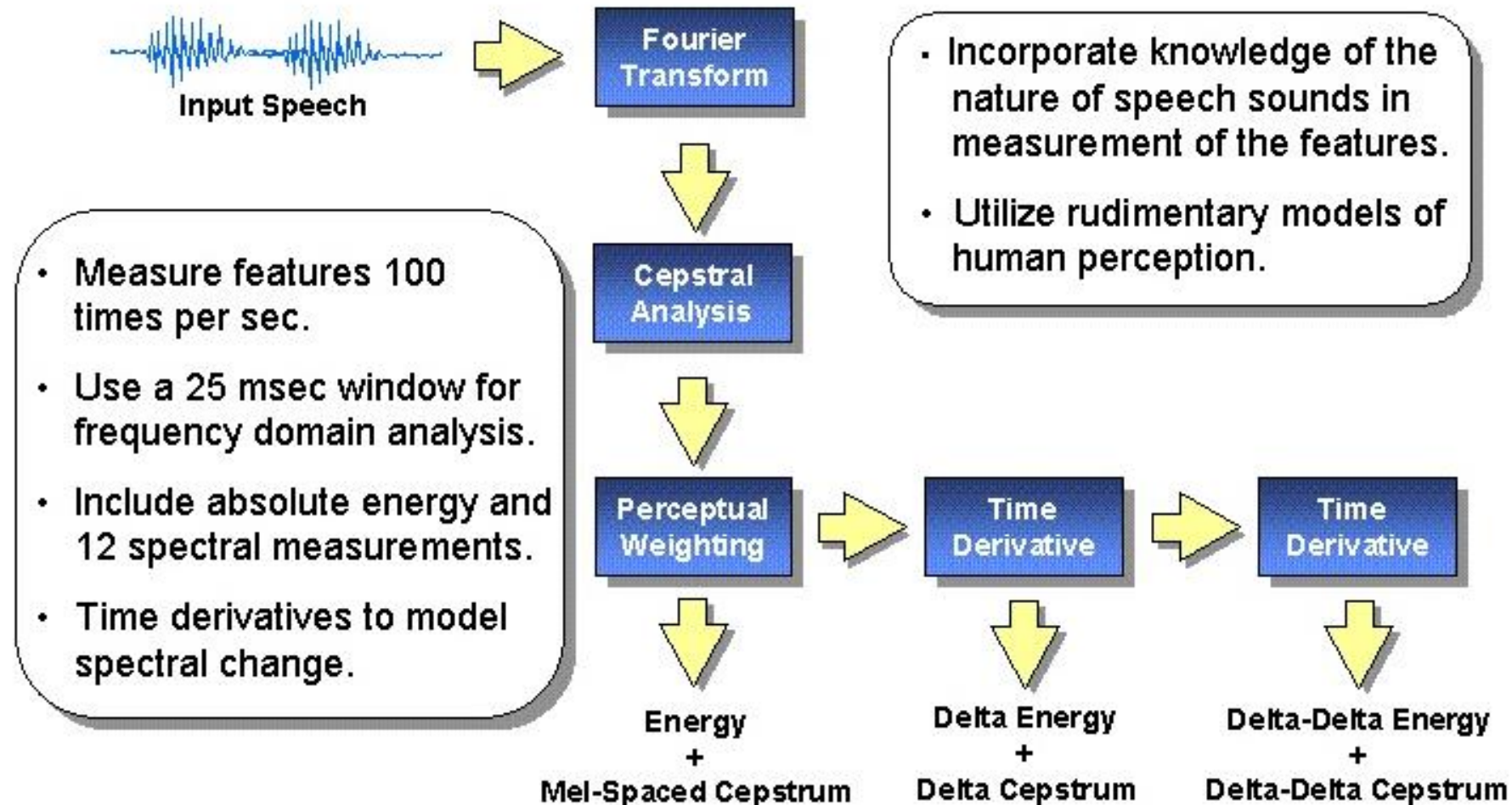
- Which of these images are most scenic?



- How can we develop a system to automatically determine scenic beauty? (Hint: feature combination)
- Solutions to such problems require good feature extraction and good decision theory.

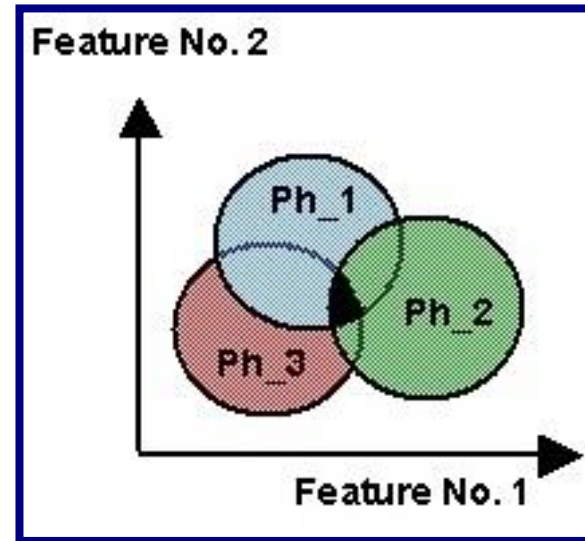
Örüntü Tanıma

Feature Extraction

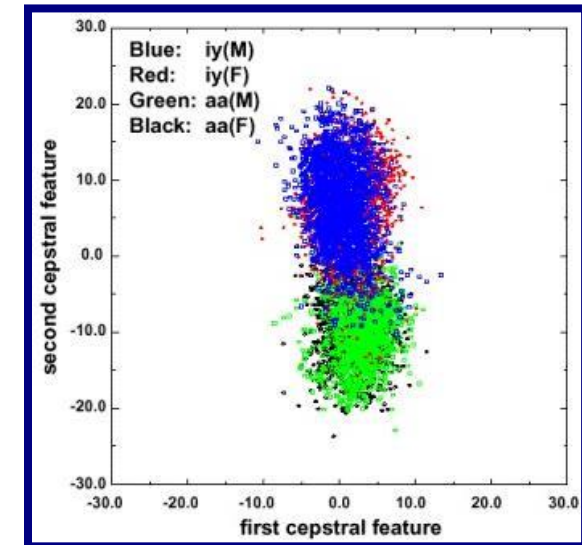


Örüntü Tanıma

Features Are Confusable



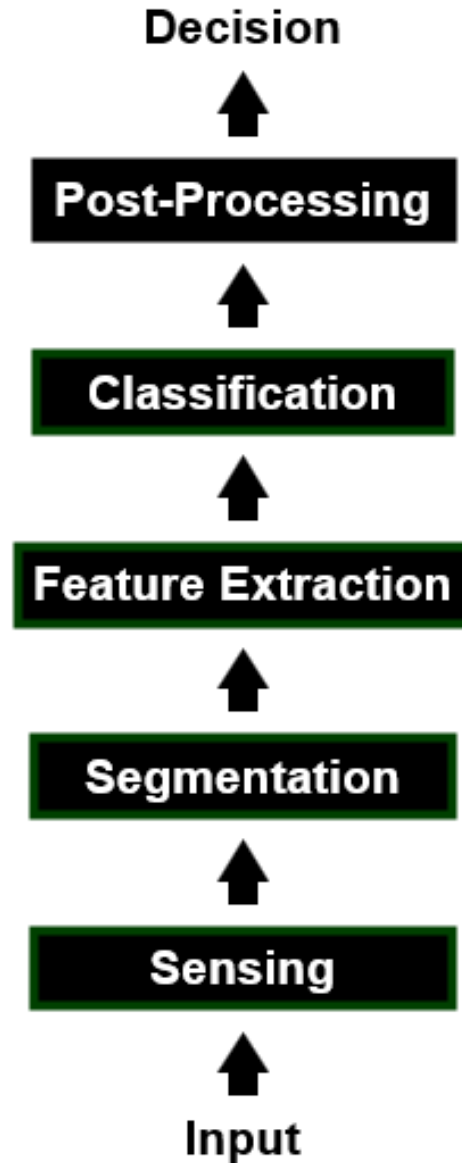
- Regions of overlap represent the classification error
- Error rates can be computed with knowledge of the joint probability distributions (see [OCW-MIT-6-450Fall-2006](#)).
- Context is used to reduce overlap.



- In real problems, features are confusable and represent actual variation in the data.
- The traditional role of the signal processing engineer has been to develop better features (e.g., “invariants”).

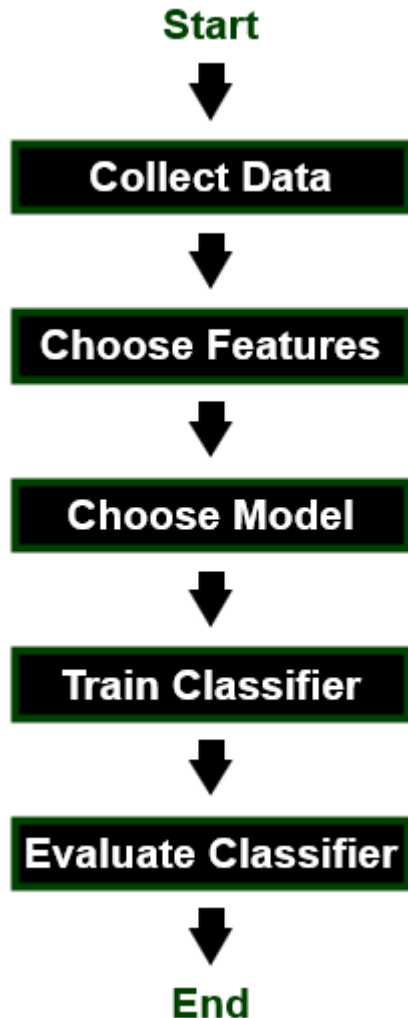
Örüntü Tanıma

Decomposition



Örüntü Tanıma

The Design Cycle



Key issues:

- “There is no data like more data.”
- Perceptually-meaningful features?
- How do we find the best model?
- How do we estimate parameters?
- How do we evaluate performance?

Goal of the course:

- Introduce you to mathematically rigorous ways to train and evaluate models.

Örüntü Tanıma

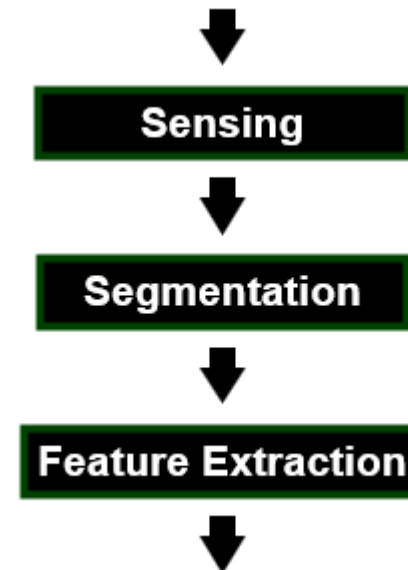
Common Mistakes

- I got 100% accuracy on...
 - Almost any algorithm works some of the time, but few real-world problems have ever been completely solved.
 - Training on the evaluation data is forbidden.
 - Once you use evaluation data, you should discard it.
- My algorithm is better because...
 - Statistical significance and experimental design play a big role in determining the validity of a result.
 - There is always some probability a random choice of an algorithm will produce a better result.
- Hence, in this course, we will also learn how to evaluate algorithms.

Örüntü Tanıma

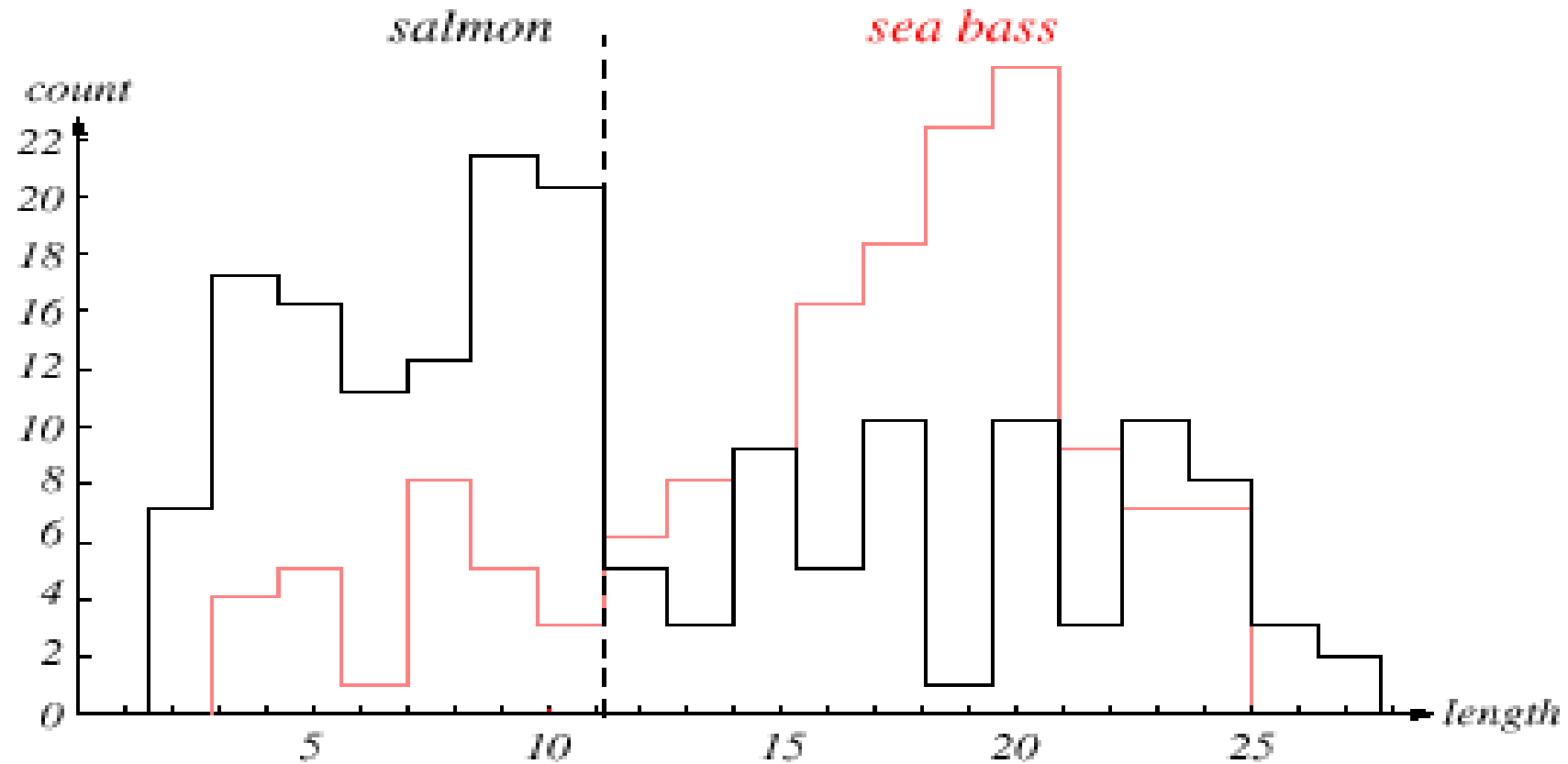
Image Processing Example

- **Sorting Fish:** incoming fish are sorted according to species using optical sensing (sea bass or salmon?)
- **Problem Analysis:**
 - set up a camera and take some sample images to extract features
 - Consider features such as length, lightness, width, number and shape of fins, position of mouth, etc.



Örüntü Tanıma

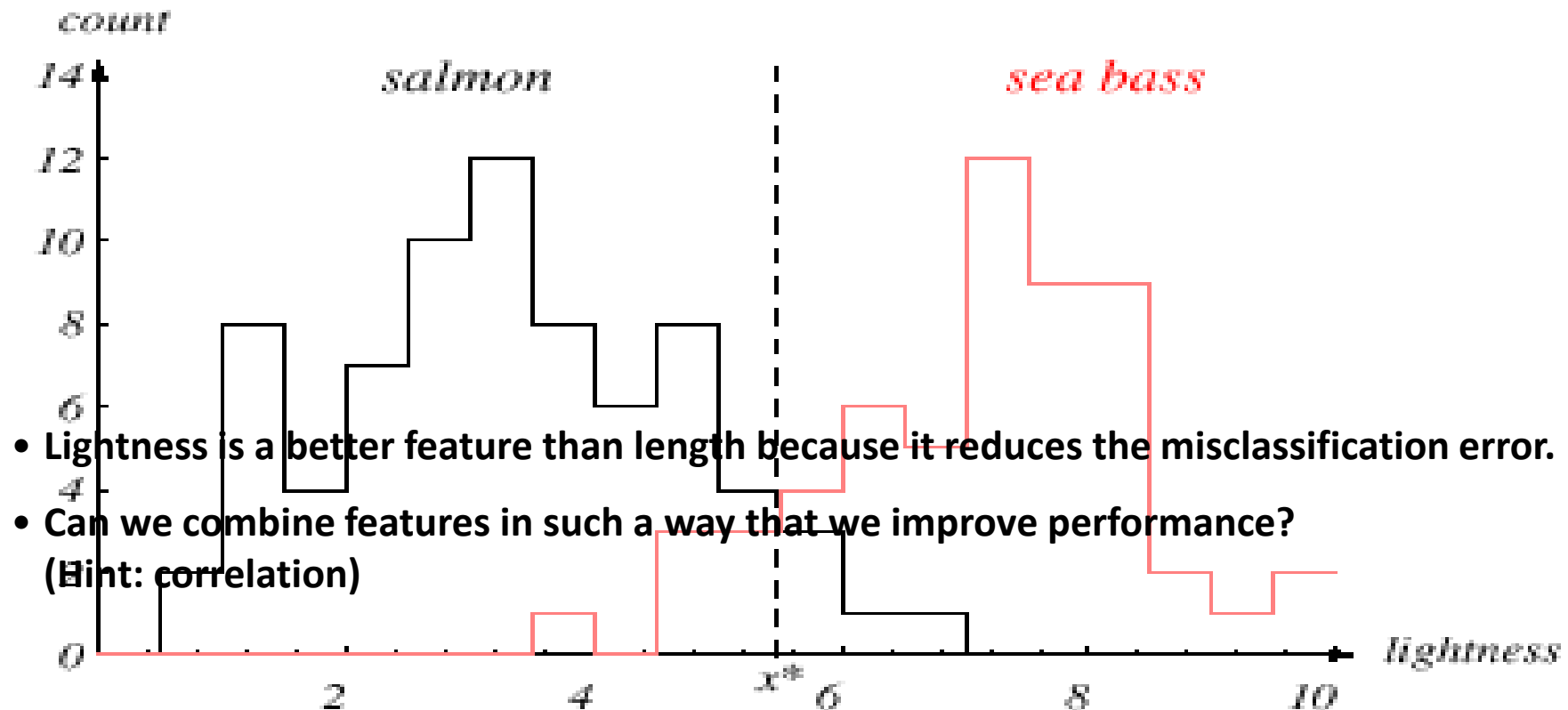
Length As A Discriminator



- Conclusion: Length is a poor discriminator

Örüntü Tanıma

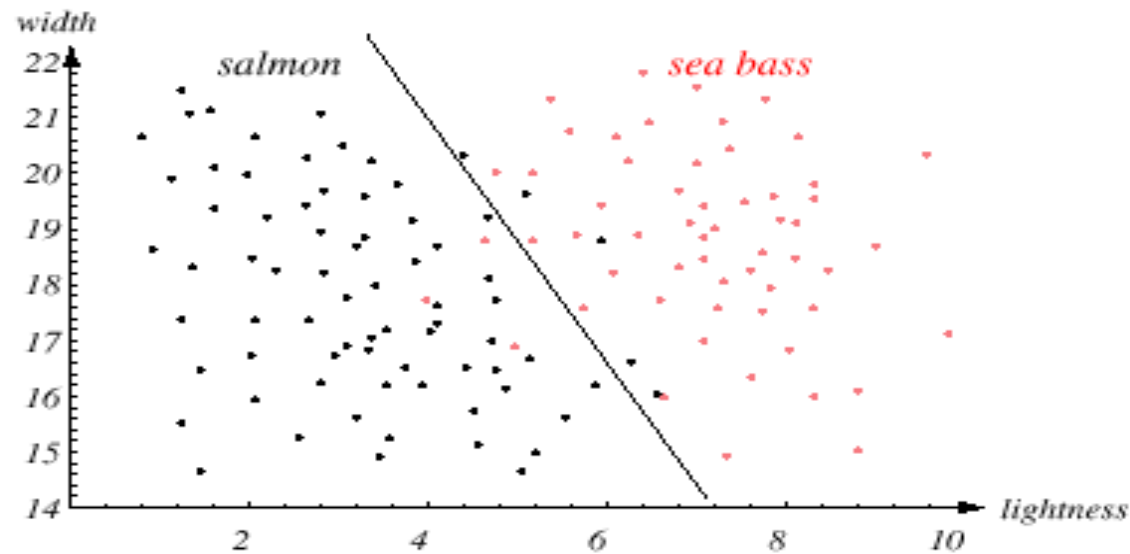
Add Another Feature



Örüntü Tanıma

Width And Lightness

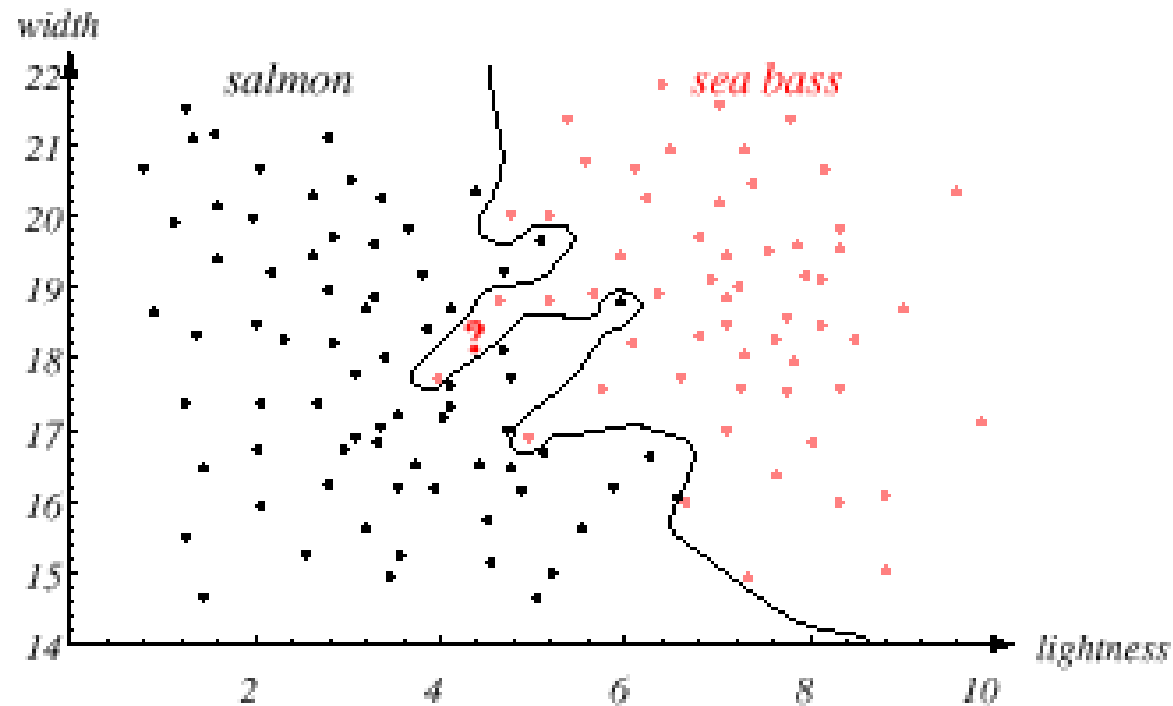
- Treat features as a N-tuple (two-dimensional vector)
- Create a scatter plot
- Draw a line (regression) separating the two classes



Örüntü Tanıma

Decision Theory

- Can we do better than a linear classifier?

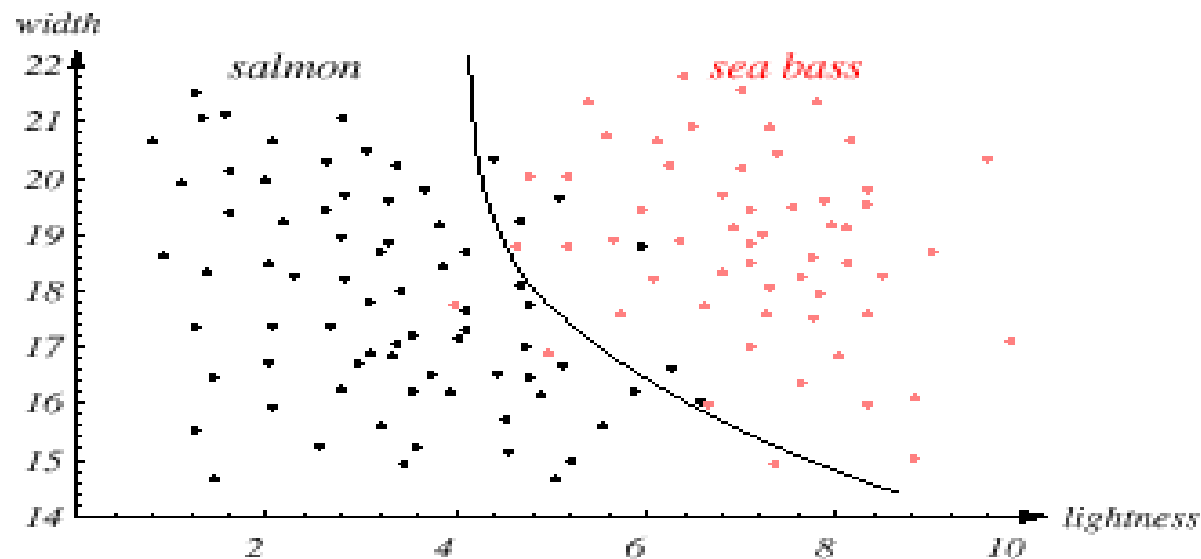


- What is wrong with this decision surface?
(Hint: generalization)

Örüntü Tanıma

Generalization and Risk

- Why might a smoother decision surface be a better choice?
(Hint: Occam's Razor).

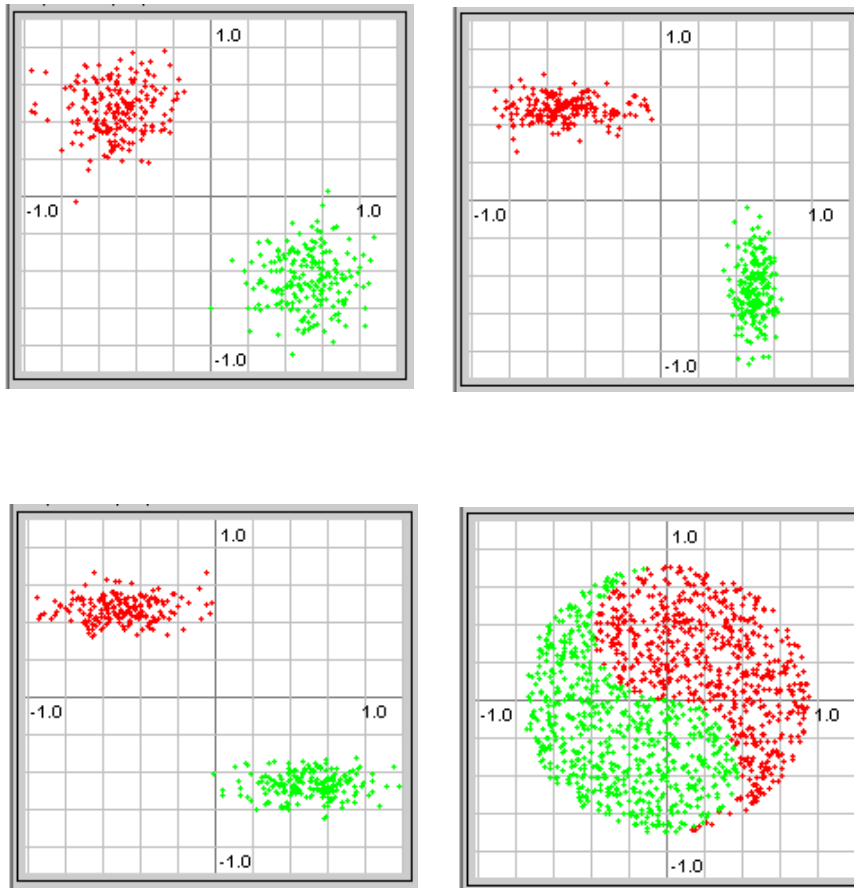


- This course investigates how to find such “optimal” decision surfaces and how to provide system designers with the tools to make intelligent trade-offs.

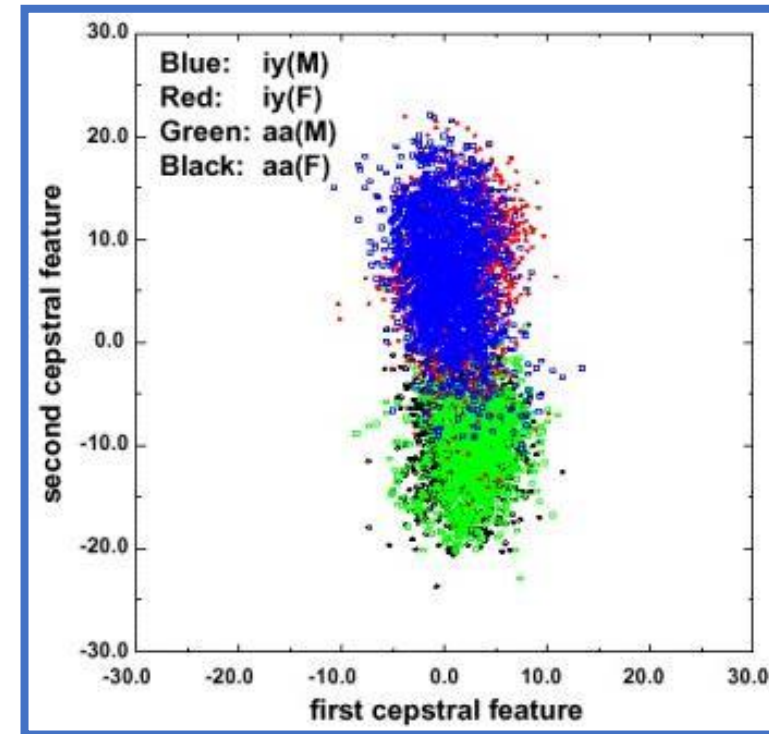
Örüntü Tanıma

Correlation

- Degrees of difficulty:



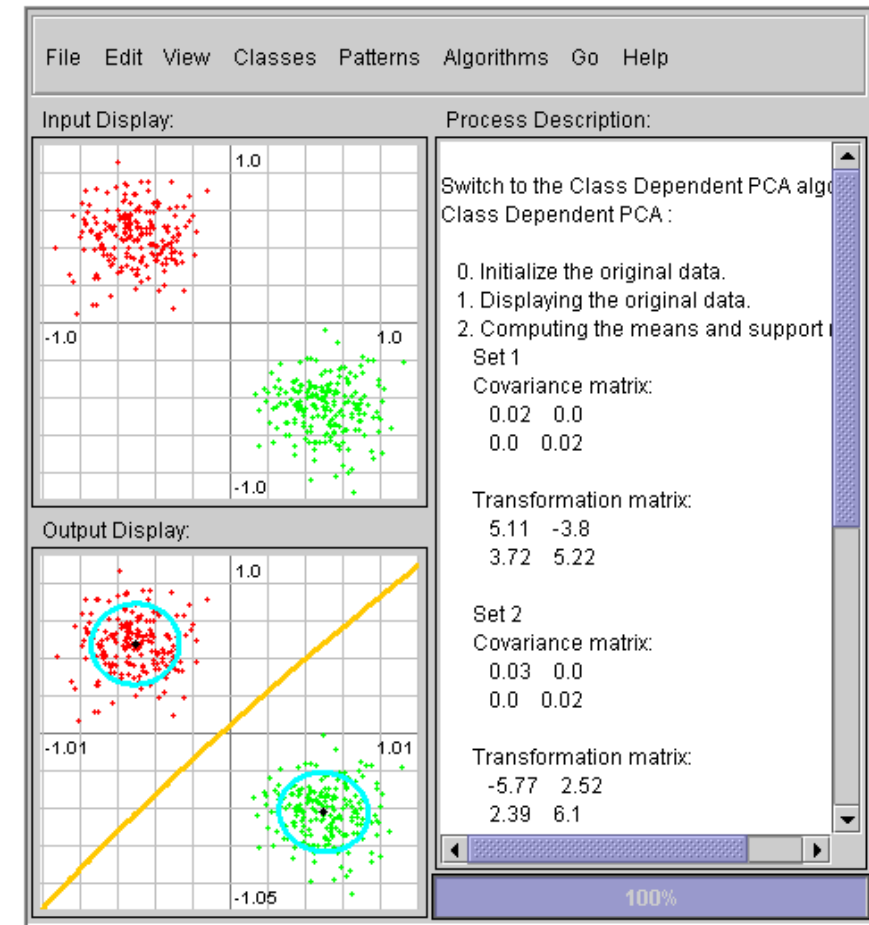
- Real data is often much harder:



Örüntü Tanıma

First Principle

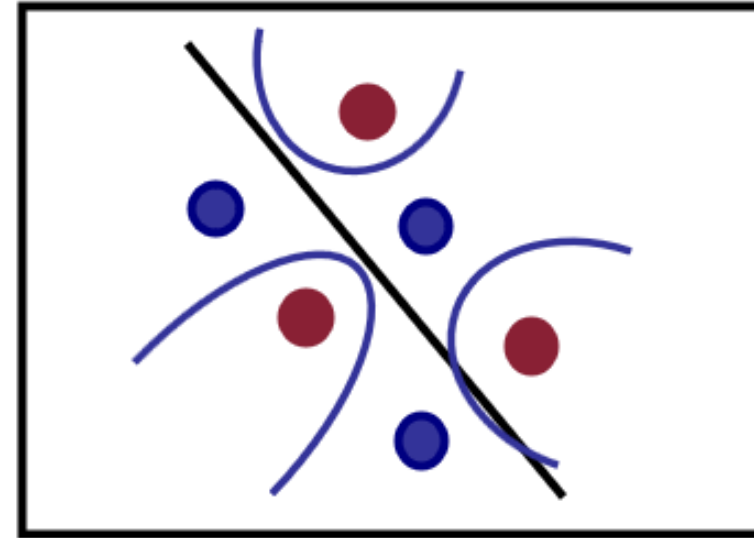
- There are many excellent resources on the Internet that demonstrate pattern recognition concepts.
- There are many MATLAB toolboxes that implement state of the art algorithms.
- One such resource is a Java Applet that lets you quickly explore how a variety of algorithms process the same data.
- An important first principle is:
 - There are no magic equations or algorithms.
 - You must understand the properties of your data and what *a priori* knowledge you can bring to bear on the problem.



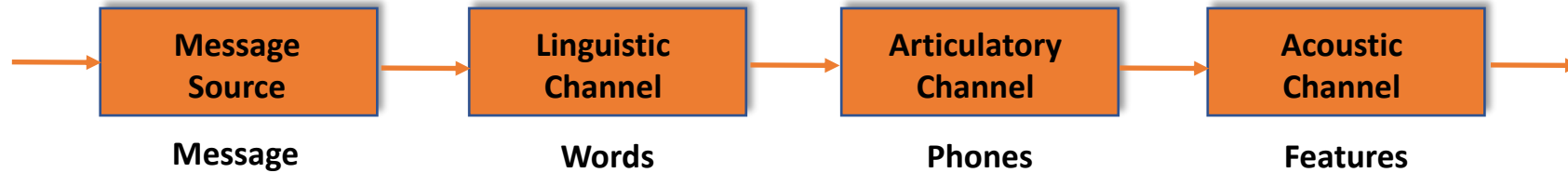
Örüntü Tanıma

Generalization And Risk

- How much can we trust isolated data points?
- Optimal decision surface is a line
- Optimal decision surface still a line
- Optimal decision surface changes abruptly
- Can we integrate prior knowledge about data, confidence, or willingness to take risk?



Örüntü Tanıma



- **Bayesian formulation for speech recognition:**

$$P(W | A) = \frac{P(A | W)P(W)}{P(A)}$$

- **Objective:** minimize the word error rate by maximizing $P(W | A)$
- **Approach:** maximize $P(A | W)$ (training)
 - $P(A | W)$: acoustic model (hidden Markov models, Gaussian mixtures, etc.)
 - $P(W)$: language model (finite state machines, N-grams)
 - $P(A)$: acoustics (ignored during maximization)
- **Bayes Rule** allows us to convert the problem of estimating an unknown posterior probability to a process in which we can postulate a model, collect data under controlled conditions, and estimate the parameters of the model.

