The Adaptive Data Cube for Integrated Sensing and Processing

Carey E. Priebe  
Johns Hopkins University

David J. Marchette  
Naval Surface Warfare Center

DARPA ISP Kickoff Meeting  
September 18-19, 2002  
Annapolis, MD
This project addresses the question of

“how statistical methods for data and dimensionality reduction should enter into data acquisition, processing and evaluation”

via an “adaptive data cube” methodology for realizing ISP integration gains from joint optimization of adaptive sensing and statistical pattern recognition.
Our general theme is that

“joint optimization of exploitation and sensing/processing is accomplished by exploitation-driven data-adaptive partitioning of initial high-dimensional measurement space $S_t, \ldots$”
Our general theme is that

“joint optimization of exploitation and sensing/processing
is accomplished by exploitation-driven data-adaptive partitioning
of initial high-dimensional measurement space $S_t, \ldots$ ”
Our general theme is that

“joint optimization of exploitation and sensing/processing is accomplished by exploitation-driven data-adaptive partitioning of initial high-dimensional measurement space $S_t$, . . . ”

“. . . and subsequent mapping of partition cells $C_j$ to new sensor settings $s_{t+1,j}$.”
Our driving philosophy is that

“recent and forthcoming advances in computational geometry will make proximity-based partitioning methodologies the choice for high-dimensional pattern recognition in the coming decades.”
Our driving philosophy is that

“recent and forthcoming advances in computational geometry will make proximity-based partitioning methodologies the choice for high-dimensional pattern recognition in the coming decades.”

Data-adaptive proximity-based partitioning of high-dimensional space is not easy, but we can now do it productively in many cases. Furthermore, when it can be done, it provides the best [says me] approach to ISP for SPR (statistical pattern recognition).
Our driving philosophy is that

“recent and forthcoming advances in computational geometry will make proximity-based partitioning methodologies the choice for high-dimensional pattern recognition in the coming decades.”

Data-adaptive proximity-based partitioning of high-dimensional space is not easy, but we can now do it productively in many cases. Furthermore, when it can be done, it provides the best [says me] approach to ISP for SPR (statistical pattern recognition).

Our method: class cover catch digraphs.
“The wealth of your practical experience with sane and interesting problems will give to mathematics a new direction and a new impetus.”
— Leopold Kronecker to Hermann von Helmholtz
“The wealth of your practical experience with sane and interesting problems will give to mathematics a new direction and a new impetus.”

— Leopold Kronecker to Hermann von Helmholtz

- artificial nose chemical sensor
- hyperspectral imagery
Applications

“The wealth of your practical experience with sane and interesting problems will give to mathematics a new direction and a new impetus.”
— Leopold Kronecker to Hermann von Helmholtz

- artificial nose chemical sensor
- hyperspectral imagery
- face detection
Applications

“The wealth of your practical experience with sane and interesting problems will give to mathematics a new direction and a new impetus.”
— Leopold Kronecker to Hermann von Helmholtz

- artificial nose chemical sensor
- hyperspectral imagery
- face detection — Diego Socolinsky
“The wealth of your practical experience with sane and interesting problems will give to mathematics a new direction and a new impetus.”

— Leopold Kronecker to Hermann von Helmholtz

- artificial nose chemical sensor
- hyperspectral imagery
- face detection — Diego Socolinsky
- computer security
Applications

“The wealth of your practical experience with sane and interesting problems will give to mathematics a new direction and a new impetus.”

— Leopold Kronecker to Hermann von Helmholtz

- artificial nose chemical sensor
- hyperspectral imagery

- face detection – Diego Socolinsky

- computer security – David Marchette
“The wealth of your practical experience with sane and interesting problems will give to mathematics a new direction and a new impetus.”
— Leopold Kronecker to Hermann von Helmholtz

- artificial nose chemical sensor
- hyperspectral imagery
- face detection — Diego Socolinsky
- computer security — David Marchette
- dna microarrays
- astronomy – sloan digital sky survey
- text document "data mining"
- functional brain imagery
Outline

- Introduction
- General Example
- Hyperspectral Example
- “Artificial Nose” Example
- A Methodology
- Face Detection — Diego Socolinsky
- Backscatter as ISP — David Marchette
- ??? Surprise: RDPs!
- Conclusion
When the exploitation task is classification, we are embarked upon the search for a theory of partitioning measurement space $S_t$ (obtained via sensor settings $s_t$), based on training data, so that observation $Z$ sensed with settings $s_t$ and falling into partition cell $C_j(s_t(Z)) \subset S_t$ will trigger new sensor settings $s_{t+1,j}$, thereby providing for superior classification performance.
General Example - I

When the exploitation task is classification, we are embarked upon the search for a theory of partitioning measurement space $S_t$ (obtained via sensor settings $s_t$), based on training data, so that observation $Z$ sensed with settings $s_t$ and falling into partition cell $C_j(s_t(Z)) \subset S_t$ will trigger new sensor settings $s_{t+1,j}$, thereby providing for superior classification performance.

Thus the goal is to obtain the original partition, and to identify the map $M$ from partition cells to new sensor settings such that, for an unlabeled observation $Z$ falling in partition cell $C_j$, the new sensor settings indicated by $M(C_j)$ are appropriate for classifying $Z$. 
When the exploitation task is classification, we are embarked upon the search for a theory of partitioning measurement space $S_t$ (obtained via sensor settings $s_t$), based on training data, so that observation $Z$ sensed with settings $s_t$ and falling into partition cell $C_j(s_t(Z)) \subset S_t$ will trigger new sensor settings $s_{t+1,j}$, thereby providing for superior classification performance.

Thus the goal is to obtain the original partition, and to identify the map $M$ from partition cells to new sensor settings such that, for an unlabeled observation $Z$ falling in partition cell $C_j$, the new sensor settings indicated by $M(C_j)$ are appropriate for classifying $Z$.

This yields an ISP-tree classification methodology – the ‘adaptive data cube’ – in which the ultimate measurement space for $Z$ is arrived at based on the exploitation task and the training data, as well as that which is observed in previous measurement space(s).
Given an initial sensor space $S_1$ (obtained via initial sensor settings $s_1$), partition $S_1$ based on training data into regions for which the best choice for $S_2$ (and $s_2$) – the best new sensor settings for resolving the unknown class label – depend on the partition cell $C_j(s_1(Z))$ into which the unlabeled observation $Z$ falls based on that first sensor reading. Each cell $C_j \subset S_1$ gives rise to a different best choice for $S_2$!

The data cube adapts ...
Hyperspectral Imagery

- HyMap hyperspectral imager:

- (one band of a) 120 band hyperspectral image:

- 7 classes: runway, pine, oak, grass, water, brush, swamp.
Hyperspectral Imagery Experiment

- Data consists of pixels from 7 classes: runway, pine, oak, grass, water, brush, swamp.
- Pixels are classed separately, using no spatial information.
- Sensor collects two “bands” which consist of Gaussian filters applied in the spectral domain. So $S_t = \mathbb{R}^2$, and $s_t$ are the means and variances of the two Gaussian filters.
- For each pixel, a 7-nearest neighbor classifier is run and top 3 classes are returned.
- Based on this, a 3-class nearest neighbor classifier is constructed, using two new Gaussian filters.
Different filters (features) are used depending on an initial classification into the top 3 classes.
Hyperspectral Partition of $S_1$
Hyperspectral Imagery Results

- Two filters selected so that the 7-nearest neighbor classifier puts the correct class into the top three most often.

- 7-nearest neighbor classifier places true class into top three 94.15%. So a lower bound on the error is 0.0585.

- Nearest neighbor classifier on original features is 0.1844.

- Using the optimal features for the nearest neighbor classifier reduces this to 0.1652.

- Two stage classifier has an error of 0.1014.
Tufts “Artificial Nose”

- Optical fibers doped with a solvatochromic dye.
- Reaction of polymer matrix with analyte produces fluorescence.
- Each fiber response is obtained by sampling at two wavelengths.
Artificial Nose Dataset

- Each (functional) observation is a multivariate time series: 37 fibers, each at two wavelengths, sampled at 60 points.
- Each (sampled) function is smoothed.
- The data set consists of 352 observations of various chemicals, and 760 observations of TCE in the confusor chemicals.
- The task is to determine whether TCE is present.
- The data set is split into equal training and testing sets.
Artificial Nose Experiment

- Select the 5 fibers resulting in the best linear classifier (resubstitution on the training set).
  This is sensor setting $s_1$.

- Training:
  - Compute the cluster catch digraph dominating set $\{X_1, \ldots, X_J\}$ for the training set.
  - For each partition cell $C_j$, identify the 5 fibers $(s_{2,j})$ resulting in the best linear classifier.

- Testing:
  - $j^* = \arg\min_j d_j(s_1(Z), s_1(X_j))$ is the index of the partition cell in $S_1$ into which $Z$ falls.
  - Classify $s_{2,j^*}(Z)$. 
The 5 fiber responses $s_1(Z)$ are collected and examined via the $ccd$ partition.
The partition cell determines a new set of fibers $s_{2,j}$ to sense.

Collect fibers 1 7 13 24 32
The new fiber responses $s_{2,j}(Z)$ are then used to classify the observation.
Other observations result in other fiber sets.

Nose Cartoon
Artificial Nose Experiment Results

- Error rate on the test set:
  - 1-nearest neighbor classifier (all fibers) 0.16
  - linear classifier (all fibers) 0.25
  - linear classifier (5 fibers) 0.28
  - ISP classifier 0.22

- 0.22 < 0.28 at the 0.05 level (McNemar’s test).
# Artificial Nose Sensor Settings

Global Features $s_1$

| 1 | 4 | 10 | 16 | 22 |

Local Features $s_{2,j}$

| 10 | 12 | 13 | 28 | 32 |
| 2  | 5  | 20 | 23 | 25 |
| 2  | 9  | 18 | 20 | 22 |
| 2  | 10 | 16 | 20 | 28 |
| 1  | 5  | 13 | 17 | 23 |
| 1  | 4  | 10 | 16 | 22 |
| 5  | 9  | 11 | 17 | 37 |
| 1  | 2  | 16 | 20 | 24 |
| 5  | 11 | 15 | 22 | 31 |
| 1  | 3  | 5  | 22 | 23 |
| 5  | 11 | 12 | 14 | 36 |
| 1  | 7  | 10 | 32 | 36 |
| 2  | 5  | 16 | 18 | 35 |
Artificial Nose Experiment Discussion

- All fibers except 21, 26, and 29 were used in the ISP-tree classifier, while no more than 5 were ever used at any stage.
- Using only 5 fibers at a time lengthens the lifetime of the sensor and may be important if there are throughput considerations.
- The ISP \textit{ccd} “local linear” classifier outperformed the other linear classifiers.
- Local nonparametric classifiers (such as the \textit{cccd}) can be used as well.
- This demonstrates a proof of concept of the ISP idea.
Artificial Nose: Future Work

Nose Day 2, March 6, 2002

Designed new data set specifically to investigate ISP issues . . .
Our approach to partitioning (high-dimensional!) sensor space goes by the acronym \textit{cccd} (class cover catch digraphs) in the supervised case and \textit{ccd} (cluster catch digraphs) in the unsupervised case.
The Class Cover Problem (CCP)

Given: $(\Omega, d)$ (a metric or dissimilarity space) with $X_1, X_2 \subset \Omega$. Let $X_1$ be the target class.

Goal: Find the smallest set of “balls”, centered at points in $X_1$, such that every point in $X_1$ is in at least one of the balls and no point in $X_2$ is in any ball.
The Class Cover Problem (CCP)

Given: \((\Omega, d)\) (a metric or dissimilarity space) with \(\mathcal{X}_1, \mathcal{X}_2 \subset \Omega\).
Let \(\mathcal{X}_1\) be the target class.
Goal: Find the smallest set of “balls”, centered at points in \(\mathcal{X}_1\), such that every point in \(\mathcal{X}_1\) is in at least one of the balls and no point in \(\mathcal{X}_2\) is in any ball.
The Class Cover Problem (CCP)

Given: \((\Omega, d)\) (a metric or dissimilarity space) with \(\mathcal{X}_1, \mathcal{X}_2 \subset \Omega\).

Let \(\mathcal{X}_1\) be the target class.

Goal: Find the smallest set of “balls”, centered at points in \(\mathcal{X}_1\), such that every point in \(\mathcal{X}_1\) is in at least one of the balls and no point in \(\mathcal{X}_2\) is in any ball.
The Class Cover Problem (CCP)

Given: \((\Omega, d)\) (a metric or dissimilarity space) with \(\mathcal{X}_1, \mathcal{X}_2 \subset \Omega\).

Let \(\mathcal{X}_1\) be the target class.

Goal: Find the smallest set of “balls”, centered at points in \(\mathcal{X}_1\), such that every point in \(\mathcal{X}_1\) is in at least one of the balls and no point in \(\mathcal{X}_2\) is in any ball.
The Class Cover Problem (CCP)

- Note that the region the balls cover is an estimate of the support of $X_1$ (assuming that it is disjoint from that of $X_2$).

- Also, it is redundant, and possibly larger than it needs to be. We wish to reduce the complexity of this cover.
Catch Digraphs

Given a collection of sets \( \{S_1, S_2, \ldots, S_n\} \) with associated “base” points \( \{t_1, t_2, \ldots, t_n\} \), we form the catch digraph \( D \) with \( V = \{v_1, v_2, \ldots, v_n\} \) and a directed edge from \( v_i \) to \( v_j \) iff \( t_j \in S_i \).
Catch Digraphs

Given a collection of sets \( \{S_1, S_2, \ldots, S_n\} \) with associated “base” points \( \{t_1, t_2, \ldots, t_n\} \), we form the catch digraph \( D \) with \( V = \{v_1, v_2, \ldots, v_n\} \) and a directed edge from \( v_i \) to \( v_j \) iff \( t_j \in S_i \).

A sphere digraph is a catch digraph where the sets are spheres and the base point for each sphere is its center.
Catch Digraphs

Given a collection of sets \( \{S_1, S_2, \ldots, S_n\} \) with associated “base” points \( \{t_1, t_2, \ldots, t_n\} \), we form the catch digraph \( D \) with 
\[
V = \{v_1, v_2, \ldots, v_n\}
\]
and a directed edge from \( v_i \) to \( v_j \) iff \( t_j \in S_i \).

A sphere digraph is a catch digraph where the sets are spheres and the base point for each sphere is its center.
Class Cover Catch Digraphs ($cccd$)

For any sets $X, Y \subseteq \Omega$ we can define the class cover catch digraph to be the catch digraph formed from the sets $B_i = \{z \in \Omega : d(z, x_i) < r_i\}$ and associated base points $x_i \in X$. 
For any sets $X, Y \subset \Omega$ we can define the class cover catch digraph to be the catch digraph formed from the sets $B_i = \{z \in \Omega : d(z, x_i) < r_i\}$ and associated base points $x_i \in X$. 
For any sets $X, Y \subset \Omega$ we can define the class cover catch digraph to be the catch digraph formed from the sets $B_i = \{ z \in \Omega : d(z, x_i) < r_i \}$ and associated base points $x_i \in X$. 
For any sets $X, Y \subset \Omega$ we can define the class cover catch digraph to be the catch digraph formed from the sets $B_i = \{z \in \Omega : d(z, x_i) < r_i\}$ and associated base points $x_i \in X$. 
Define a *dominating set*, $S$, of a digraph $D = (V, A)$ as follows: $S \subseteq V$ such that $\forall v \in V$, $v \in S$ or $\exists w \in S$ such that $(w, v) \in A$. 

![Diagram of a digraph](image-url)
Define a *dominating set*, $S$, of a digraph $D = (V, A)$ as follows: $S \subset V$ such that $\forall v \in V$, $v \in S$ or $\exists w \in S$ such that $(w, v) \in A$. 

Solution to CCP $\iff$ Minimum Size Dominating Set in $cccd$
Dominating Sets: Greedy Algorithm

Set \( D = \emptyset, \quad R = X_1 \).

While \( R \neq \emptyset \)

\[ x = \arg \max_{z \in X_1 \setminus D} |B(z, d(z, X_2)) \cap R| \]  [break ties arbitrarily]

\[ D = D \cup \{x\} \]

\[ R = R \setminus B(x, d(x, X_2)) \]

EndWhile

Return \( D \).

The greedy algorithm finds the ball that covers the most points not already covered. Tie breaking can take radius into account (resulting in smaller, denser balls in the case of a tie).
Cluster Catch Digraphs (ccd)

- How do we modify the ccd to do clustering?
- The problem is to define the algorithm that determines the radius.
- We will use a methodology that defines the radius in terms of a local density of points.
- This performs a hypothesis test against a particular alternative of “no clustering”.
Radius Computation

- Each observation computes its radius independently of the other radii. This observation is referred to as the “center point” of the resulting ball.

- A random walk is defined along the empirical distribution function of the distance of observations from the center point.

- The random walk is compared to a hypothesis of “no clustering” which is defined as the uniform distribution in the ball.

- The radius is chosen based on the deviation from the null hypothesis: a Kolmogorov-Smirnov test.
Random Walk \( ccd \)

Idea: Let each point choose the radius of its covering ball.
Idea: Let each point choose the radius of its covering ball.

Plot radius against number of points covered.
Random Walk $ccd$

Idea: Let each point choose the radius of its covering ball.

Plot radius against number of points covered.
Random Walk $ccd$

Idea: Let each point choose the radius of its covering ball.

Plot radius against number of points covered.
Random Walk \textit{ccd}

Idea: Let each point choose the radius of its covering ball.

Plot radius against number of points covered.
Random Walk \( ccd \)

Idea: Let each point choose the radius of its covering ball.

Plot radius against number of points covered.
Random Walk $ccd$

Idea: Let each point choose the radius of its covering ball.

Plot radius against number of points covered.
Random Walk $ccd$

Idea: Let each point choose the radius of its covering ball.

Plot radius against number of points covered.
Random Walk \textit{ccd}

Idea: Let each point choose the radius of its covering ball.

Plot radius against number of points covered.
Random Walk $ccd$

Idea: Let each point choose the radius of its covering ball.

Plot radius against number of points covered. We choose the radius according to:

$$r^* = \arg \max_{r \geq 0} \{RW(r) - mr^d\}$$

This is a K-S test against the null hypothesis that the data is uniform.
Radius Computation Details

- Let \( F(r) \) be the empirical distribution function for distance from the center point, and \( F_0 \) the distribution function under the null hypothesis.
- Define

\[
\begin{align*}
l(r) &= \min_{q \leq r} F(q) - F_0(q) \\
u(r) &= \max_{q \leq r} F(q) - F_0(q).
\end{align*}
\]

- Let \( k_\alpha \) be the value of the K-S statistic at which one would reject with level \( \alpha \).
- Let \( R \) be the first radius such that either \(-l(r) \geq k_\alpha\) or \( u(r) \geq k_\alpha\).
Radius Computation Details

- If the rejection first occurs as a result of $l(r)$, set
  \[ r_x = \arg \max_{q<r} F(q). \]

- Else if rejection occurs do to $u(r)$ continue the walk while $u(r)$ is non-decreasing, and set $r_x$ to the radius prior to the first time $u(r)$ decreases.

- Else set $r_x = \arg \max F(q)$. 

---

![Diagram showing the radius computation process](image.png)
The Null Hypothesis

- In two dimensions, the distribution function for uniform (Poisson) data is proportional to $r^2$.
- In dimension $d$ it is proportional to $r^d$.
- Due to the curse of dimensionality, and the fact that one generally believes the data lives on a lower dimensional subspace, one might choose $d' < d$.
- The selection of the constant of proportionality, related to the intensity of the process, is also a parameter to adjust.
- Both of these should be adjusted locally, since they are, in principle, local properties of the data.
The standard greedy algorithm can be employed for the selection of the dominating set in the \textit{ccd}.

Alternatively, the algorithm can be modified to be greedy on the K-S statistic:

- Select the ball with the largest K-S statistic.
- Recompute the K-S statistic for all the balls as if the covered points did not exist.

Note: there is a school of thought that the radii should also be recomputed at this point.
Acknowledgements

This research was partially supported by DARPA’s

*Human Identification at a Distance*

program (Jonathon Phillips, PM), and

*Equinox Corporation*
The Face Detection Problem

We consider one of the most challenging face detection scenarios:
We consider one of the most challenging face detection scenarios:

- Grayscale imagery.
The Face Detection Problem

We consider one of the most challenging face detection scenarios:

- Grayscale imagery.
- Static shots, i.e. no video.
The Face Detection Problem

We consider one of the most challenging face detection scenarios:

- Grayscale imagery.
- Static shots, i.e. no video.
- Multiple (or no) faces per image.
The Face Detection Problem

We consider one of the most challenging face detection scenarios:

- Grayscale imagery.
- Static shots, i.e. no video.
- Multiple (or no) faces per image.
- Faces occur at different scales.
The Face Detection Problem

We consider one of the most challenging face detection scenarios:

- Grayscale imagery.
- Static shots, i.e. no video.
- Multiple (or no) faces per image.
- Faces occur at different scales.
The Face Detection Problem

We consider one of the most challenging face detection scenarios:

- Grayscale imagery.
- Static shots, i.e. no video.
- Multiple (or no) faces per image.
- Faces occur at different scales.
Challenges, Goals and Restrictions

Number one challenge: Horrendously skewed priors: non-faces are many orders of magnitude more common than faces.

The goals for our detector are:
High detection rate.
Minimal false alarm rate.
Real-time operation!

We have to place some restrictions so we can meet our goals:
Detect only frontal faces, not profiles.
Detect only upright faces, not tilted ones.
Challenges, Goals and Restrictions

Number one challenge:

- Horrendously skewed priors: non-faces are many orders of magnitude more common than faces.

The goals for our detector are:

- High detection rate.
- Minimal false alarm rate.
- Real-time operation!

We have to place some restrictions so we can meet our goals:

- Detect only frontal faces, not profiles.
- Detect only upright faces, not tilted ones.
Challenges, Goals and Restrictions

Number one challenge:

- **Horrendously** skewed priors: non-faces are many orders of magnitude more common than faces.

The goals for our detector are:

- High detection rate.
- Minimal false alarm rate.
- **Real-time operation!**
Challenges, Goals and Restrictions

Number one challenge:

- Horrendously skewed priors: non-faces are many orders of magnitude more common than faces.

The goals for our detector are:

- High detection rate.
- Minimal false alarm rate.
- **Real-time operation!**

We have to place some restrictions so we can meet our goals:

- Detect only frontal faces, not profiles.
- Detect only upright faces, not tilted ones.
Training Data
Training Data

Normalized faces
Training Data

Normalized faces

Normalized non-faces
In order to meet the challenges of heavily skewed priors and real-time requirements, we structure our classifier as a one-sided decision tree, each leaf of which is a CCCD classifier. At each leaf of the tree, we may classify a sample as a non-face, and stop the process, or as a potential face and descend to the next level for a more refined test. This way, only a very small portion of the classifier is applied to the majority of the image.
Computationally efficient detection

Input image  Exit depth

The average exit depth for this image is 1.34, using a CCCD tree with 15 levels! The vast majority of background is correctly classified in the earliest stages.
Successive stages of the CCCD tree are **boosted** to increase performance on the background class, while maintaining an upper bound on the error rate of the face class.
Boosted Training Algorithm

**Data**: Data $\mathcal{T}_0$ and $\mathcal{T}_1^0$. Number of stages $\beta$. Error threshold $\alpha$.

Let $i = j = k = 1$, $\mathcal{X}_1^0 = \mathcal{X}_1$;

while $i \leq \beta$ do

Select $C_i^0 = \{c_{0,1}, \ldots, c_{0,f_i}\}$ and $R_i^0$ as in standard CCCD algorithm;

repeat

Select $c_{1,j}^i$ and $r_{1,j}^i$ as in standard CCCD algorithm, using $\mathcal{X}_1^{k-1}$ as training data;

Adjust the scaling factors to enforce the required empirical performance bound on the face class;

Let $\mathcal{T}_1^k \subseteq \mathcal{T}_1^0$ be the class-1 observations incorrectly classified by the current classifier, and $\mathcal{X}_1^k$ be the misclassified class-1 training observations;

until $|\mathcal{T}_1^k|/|\mathcal{T}_1^{k-1}| \geq \alpha$;

end
Boosted Tree Classification

**Data**: Number of stages $\beta$. Number of face prototypes per stage, $f_i$, and number of sub-stages per stage, $n_i$, for $i = 1, \ldots, \beta$

Let $i = 1$;

while $i \leq \beta$ do

Let $m_0 = \min_{k=1}^{f_i} \rho(x, c_{0,k}^i)$;

for $j = 1, \ldots, n_i$ do

if $\rho(x, c_{1,j}^i) < m_0$ then

Classify as non-face and exit;

end

end

end

Classify as face;
Examples
Companies report hundreds of denial of service attacks each year.

They report millions (billions?) of dollars lost.
Companies report hundreds of denial of service attacks each year.

They report millions (billions?) of dollars lost.

They lie.
Companies report hundreds of denial of service attacks each year.

They report millions (billions?) of dollars lost.

They lie.

We need a way to reliably estimate the number, type, and sizes of denial of service attacks on the Internet, without relying on self-reporting by victims. And it must be timely, not days (weeks) after the fact.
Companies report hundreds of denial of service attacks each year.
- They report millions (billions?) of dollars lost.
- They lie.

We need a way to reliably estimate the number, type, and sizes of denial of service attacks on the Internet, without relying on self-reporting by victims. And it must be timely, not days (weeks) after the fact.

This talk will describe an ISP sensor to perform this task.
Background

Typical Denial of Service Attack: Syn Flood. Attacker floods the victim with connection requests.
Victim

Attackers send spoofed SYN packets
(SYN packets are connection requests)
(“Spoofed” means they put in fake source IPs)

The Internet
Background

Victim responds with SYN/ACK packets

Attacker(s)

The Internet

Victim

ISP for SPR – September in Annapolis
Background

Victim

Sensors at the spoofed addresses see the response packets

Attacker(s)

The Internet

US
Counting the unsolicited packets allows a **realtime** estimate of the number of attacks on the Internet, **without** consulting the victims.
Why ISP?

- Network loads, bandwidth and disk storage.
- Realtime constraints.
- Need to distinguish between victims and attackers.
An ISP Sensor

- **Backscatter Detector**
- **Denial-of-Service Detector**
- **Classifier**
- **Filters**
- **Counter**
An ISP Sensor

Sensor consists of filters determining packets collected
An ISP Sensor

- Backscatter Detector
- Denial-of-Service Detector
- Classifier
- Counter

Ignore legitimate sessions
An ISP Sensor

Only consider packets from DOS

Denial-of-Service Detector

Counter

Classifier

Filters

Backscatter Detector
An ISP Sensor

- Filters
- Backscatter Detector
- Denial-of-Service Detector
- Counter
- Classifier

Different attacks/different packets
An ISP Sensor

- Filters
- Backscatter Detector
- Denial-of-Service Detector
- Classifier
- Counter

Output:
- Attack counts
- Attack class
- Packets

ISP for SPR – September in Annapolis – p.53/66
Classifier Issues

The Backscatter Detector needs to distinguish between backscatter and:

- Normal traffic.
- Miscellaneous errors.

Thus, it is basically a statefull machine.
Classifier Issues

The Backscatter Detector needs to distinguish between backscatter and:

- Normal traffic.
- Miscellaneous errors.

Thus, it is basically a statefull machine.

The Denial-of-Service Detector needs to distinguish between:

- Packets from denial of service attacks.
- Packets from scans of the victim.
- Packets from other direct attacks on the victim.
Classifier Issues

The Classifier needs to:

- Determine the type of attack.
- Determine the number of attackers.
- Determine the attack tool used.

These use different information from the packets. Some packets do not lend themselves to some of these goals.
The sensor will adjust the types of packets retained based on:

- Load on the monitored network.
- Determination that the packets are likely to be backscatter.
- Feedback from the classifier indicating the type of attack.

Different backscatter (hence different attacks) require different packets, and different attributes from the packets, to be retained.
Some Victims

![Graphs showing some victims](image-url)
Discussion

An ISP sensor for backscatter would provide a real time assessment of the Internet threat level. The intent is to develop one that:

- Detects, measures and records the DOS attacks on the Internet.
- Changes the packets collected based on feedback from the classifier.
- Determines useful information about the attacks such as:
  - The number of packets sent.
  - The number of attackers.
  - The severity (did it work?).
  - The tool used.

ISP is needed due to the huge number of packets to be gathered, processed and stored.
MCM PP as ISP

Random Disambiguation Paths

... or...

Mine CounterMeasures Path Planning as ISP
Future Combat Systems

(Photo courtesy Federation of American Scientists.)
Coastal Battlefield Reconnaissance and Analysis (COBRA) imagery

(Minefield data was provided by NSWC Coastal Systems Station, Dahlgren Division, Panama City, Florida, with support from United States Marine Corps Amphibious Warfare Technology.)
Our Goal: Minefield Traversal
\ell (p^*) = 978
The random disambiguation path $p^{**}$ has expected length

$$E [\ell (p^{**})] = 1112 \rho_i^* + 708(1 - \rho_i^*) + c < 978 = \ell (p^*)$$

for $c = 0$ and $\rho_i^* < 0.668$, for example.
Collaborators

David Marchette
Edward Wegman
David Walt
Diego Socolinsky
Jason DeVInney
Elvan Ceyhan
Kendall Giles

A dozen & change “class cover” and related manuscript citations are available at <http://www.mts.jhu.edu/~priebe/hdda1.html>. 