Planar Languages and Learnability

Alexander Clark, Christophe Costa Florêncio, Chris Watkins, Mariette Serayet

Department of Computer Science, Royal Holloway, University of London

Département Informatique, Université de St Etienne

Planar langauges: motivation

- **Efficient unsupervised learning of mildly context-sensitive grammars**

Planar languages: simple example

- Parikh map: maps a string to a vector of counts
- Parikhs theorem
  - image of a CF language is semilinear

$$\Sigma=\{a,b\}$$
$$p(abba)=(2\,2)$$
$$p(aab)=(2\,1)$$

CF language

$$L = \{ w : |w|_a = |w|_b \}$$

$$\{ ab, ba, bbaa, aaabbb, ababab \ldots \}$$

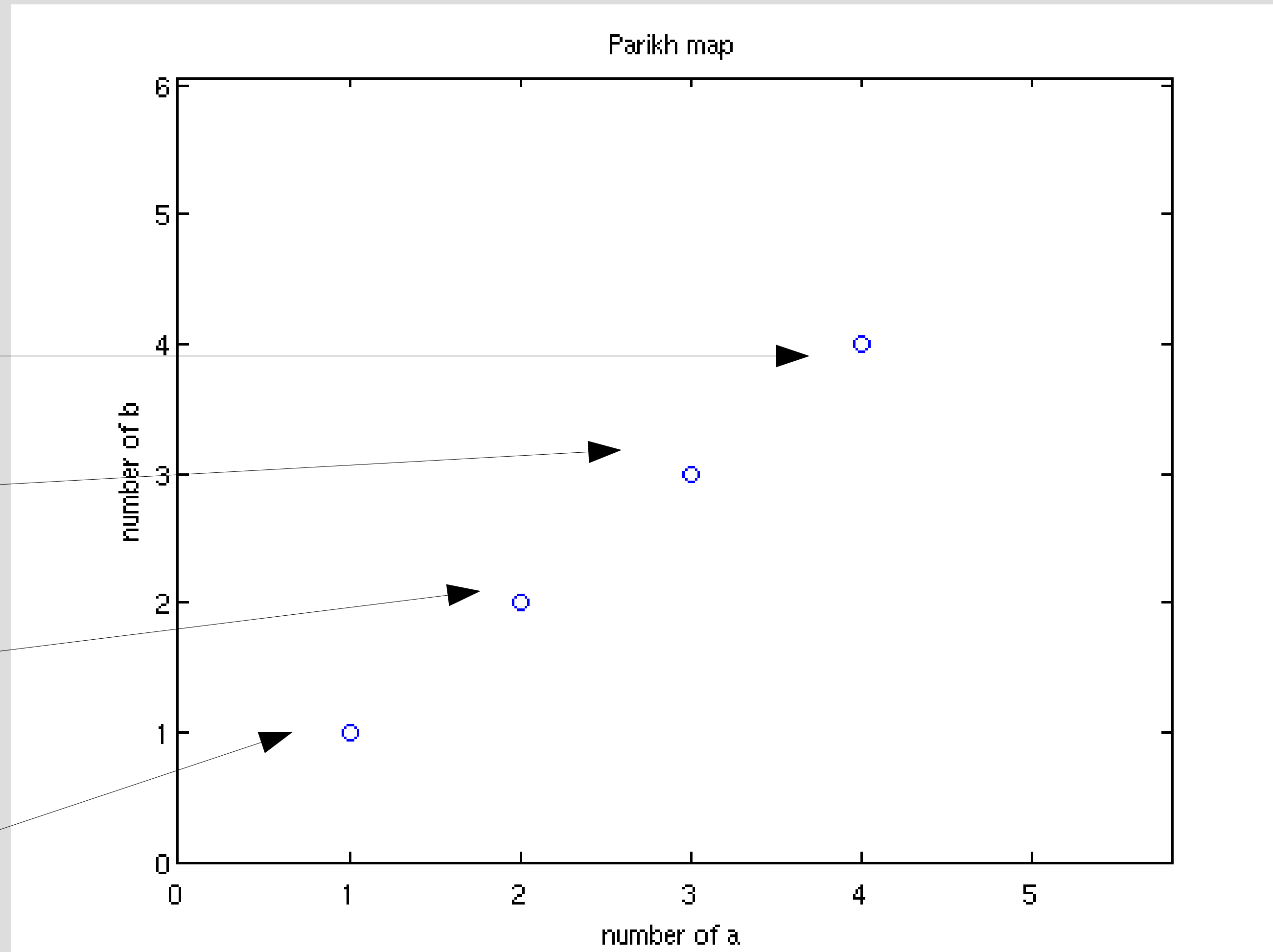
L lies on a line

baababba

bbbaaa

aabb
baba

ba
ab



We can learn languages from positive data

- Project data into feature space
- Find lowest dimensional hyperplane that contains all the data
- Given unknown string w
  - measure perpendicular distance to plane
  - if this is zero (or very small), w in L
  - if it is big, w is not in L

Outline

- What, why, wherefore of kernels
- String kernels
- Planar languages
- Theoretical results: expressive power, learnability, injectivity
- Experimental results
- Future research
- Conclusion

Kernel functions 1

- Map from data points to points in feature space
- Function must be positive semi-definite
- Can be used with any linear pattern learning algorithm

Kernel functions 2

- **Computational problem: feature space generally high/infinite dimension**
- **Solution: compute just inner product, feature vector remains implicit:**

$$K(u,v)=\langle \phi(u),\phi(v)\rangle$$

String kernels

- **Map strings to points in feature space**
- **All-*k*-subsequences, *k*-subsequences: based on counts of subsequences of length (up to) *k***
- ***k* = 1: Parikh kernel**
- **Gap-weighted: based on counts of subsequences of length (up to) *k*, weighted by λ^n, *n* = size of gap**
- ***p*-Spectrum: based on counts of contiguous subsequences of length *p***

Planar languages

- **Informally: κ-planar language is set of strings corresponding to hyperplane in κ's feature space**
- **intuitively learnable from positive data**
- **Formally:**

$$L = \{ w \in \Sigma^* : \exists \alpha_1 \ldots \alpha_n \in R,$$
$$\exists u_1 \ldots u_n \in \Sigma^* : \sum_i^n \alpha_i \phi(u_i) = \phi(w) \}$$

Expressive power

- Depends on kernel
- Generally, planar languages do not conform to Chomsky hierarchy
- All-*k*-subsequences contains non-mildly context sensitive languages
- *k*-testable languages are planar for *p*-Spectrum kernel

Closure properties

- Generally, planar languages do not fit into Chomsky hierarchy
- Not closed under:
  - concatenation, union, homomorphism...
- Closed under:
  - reversal
  - intersection

Injectivity

- ***k*-subsequences not injective: for *k* = 2, "abba" and "baab" map to same point**
- **But: length of such strings grows exponentially in *k*?**
- **Gap-weighted is injective when decay factor transcendental number**

Learnability

- Paradigm: identification in the limit
- Characteristic set has polynomial size
- Polynomial computation time
- Polynomial number of mind changes
- Learner exists that is consistent, monotone increasing and incremental

Proof of learnability

- Based on properties of learning algorithm SPAN:
  - loop over input data
  - if current datapoint does not correspond to point in hyperplane spanned by base, add to base

Finite elasticity

- In case that feature space defined by κ has finite dimension, class of κ-planar languages has finite elasticity
- True for GapWeighted(+) kernels
- Finite unions of such classes also have finite elasticity

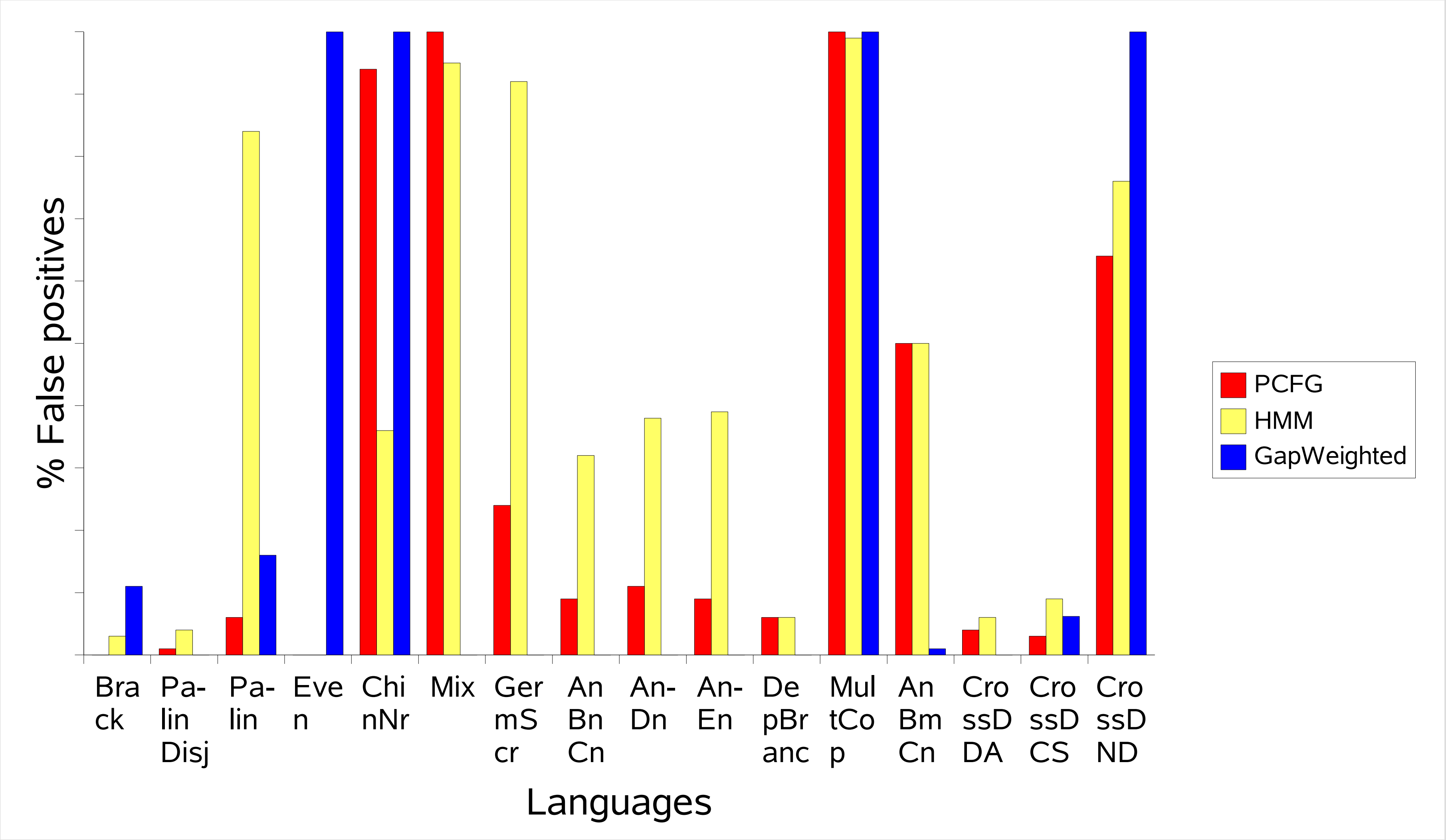
Computation

```
function handle,
% kernel hyperparameter.
% Written by Alex Clark but heavily based on
% John Shawe-Taylor and Nello Cristianini's code
% Jan 11 2006.

n = size(train,1);
for i = 1:n
  for j = i:n
    v = kernel(train{i},train{j},param);
    K(j,i) = v;
    K(i,j) = v;
  end
end

% K is the Gram matrix
D = sum(K)/n;
```

Experimental results

- Planar language learner, implemented in Matlab
  - computes eigendecomposition of translated *Gram* matrix
- Synthetic datasets:
  - palindromes
  - MIX language
  - AnBmCnDm
  - Crossing serial dependencies
  - ...

PCFG vs HMM vs GapWeighted



Future directions

- Reducing noise sensitivity
- Kernels customized for natural language
- Preimage problem
- Work with real corpora: very high dimensionality (large alphabets), high sample complexity
- Solution: projections/distribution kernels?

Conclusion

- Planar languages constitute a new approach to GI
- Inherently efficiently learnable
- Right expressive power
- Potential for use in NLP