Visual Recognition
{Generative Models}

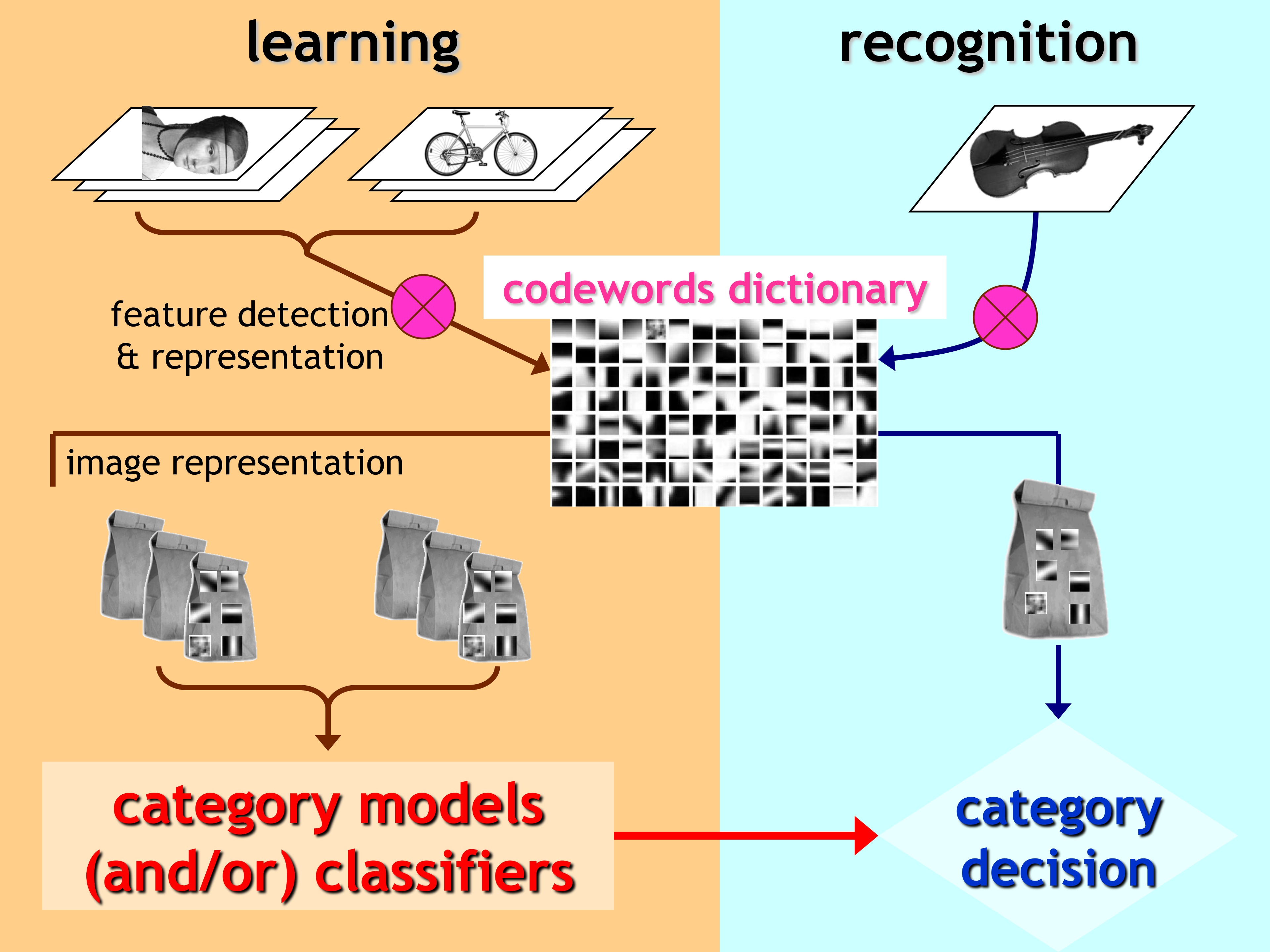
Nikhil Rasiwasia

Scientist

Yahoo! Labs Bangalore

Visual Recognition: Three Steps.

- Representation
  - How to represent images
- Learning
  - How to form the classifier, given training data
- Recognition
  - How the classifier is to be used on novel data


learning
recognition
codewords dictionary
feature detection
& representation
image representation
category models
(and/or) classifiers
category
decision

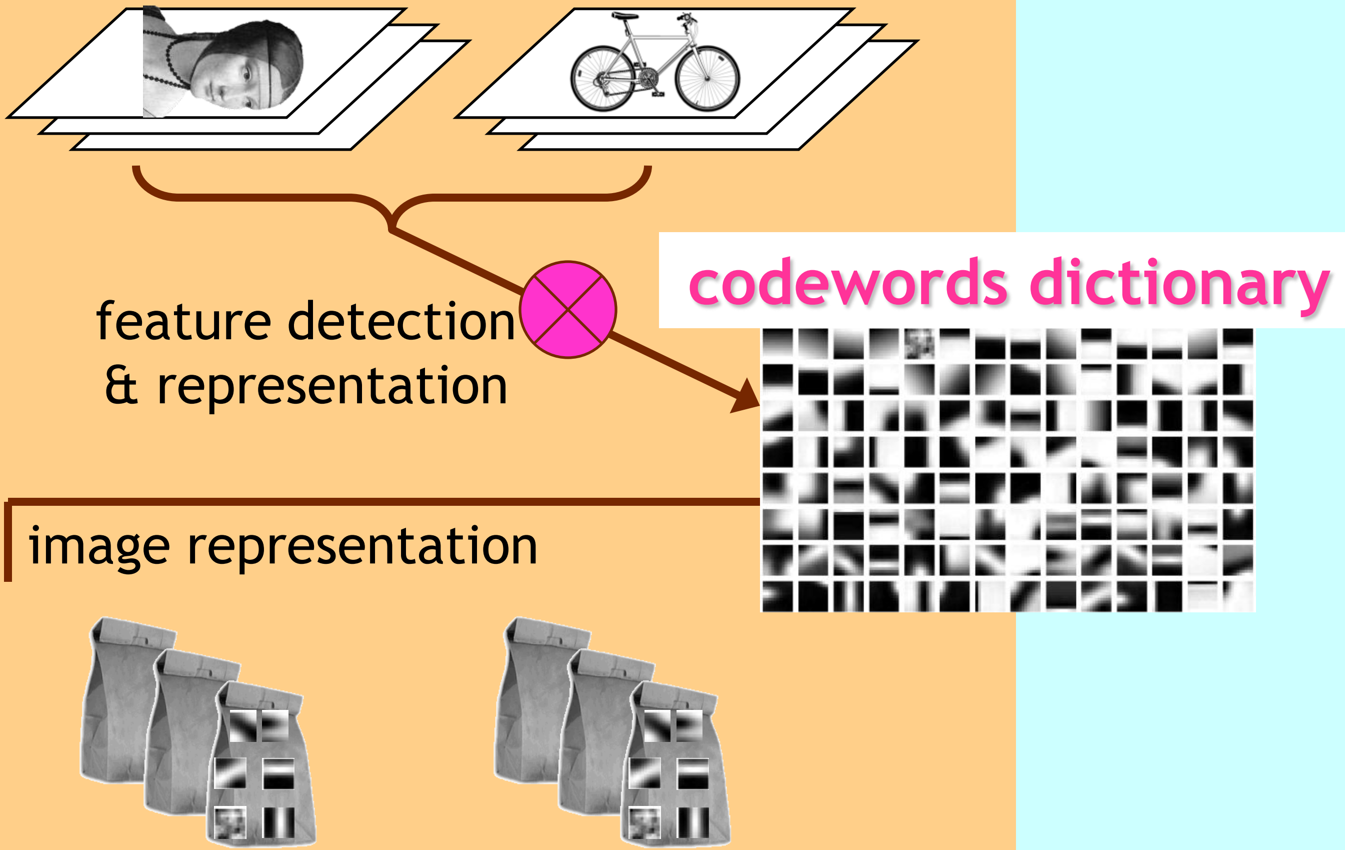
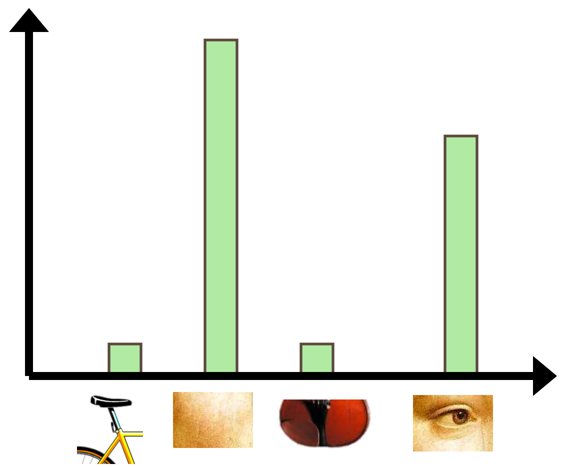
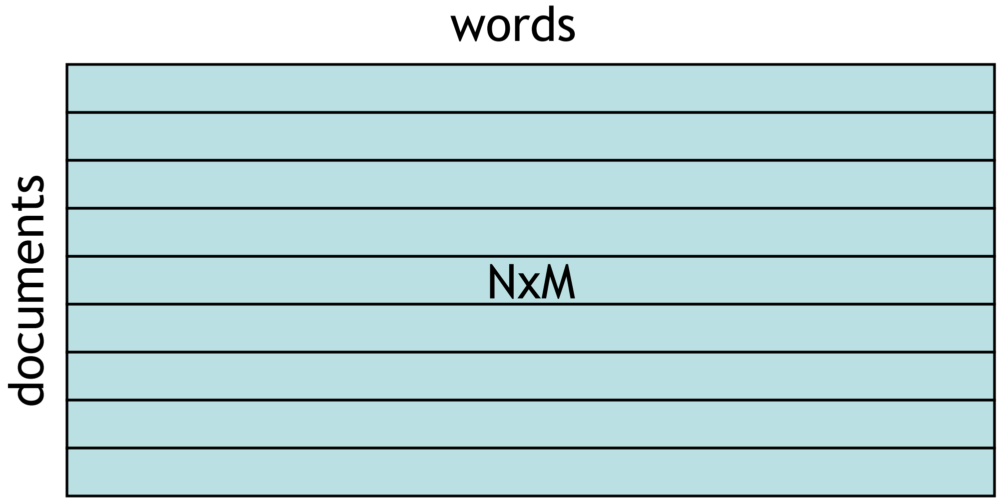

codewords dictionary
feature detection
& representation
image representation

Image → Bag of ‘words’

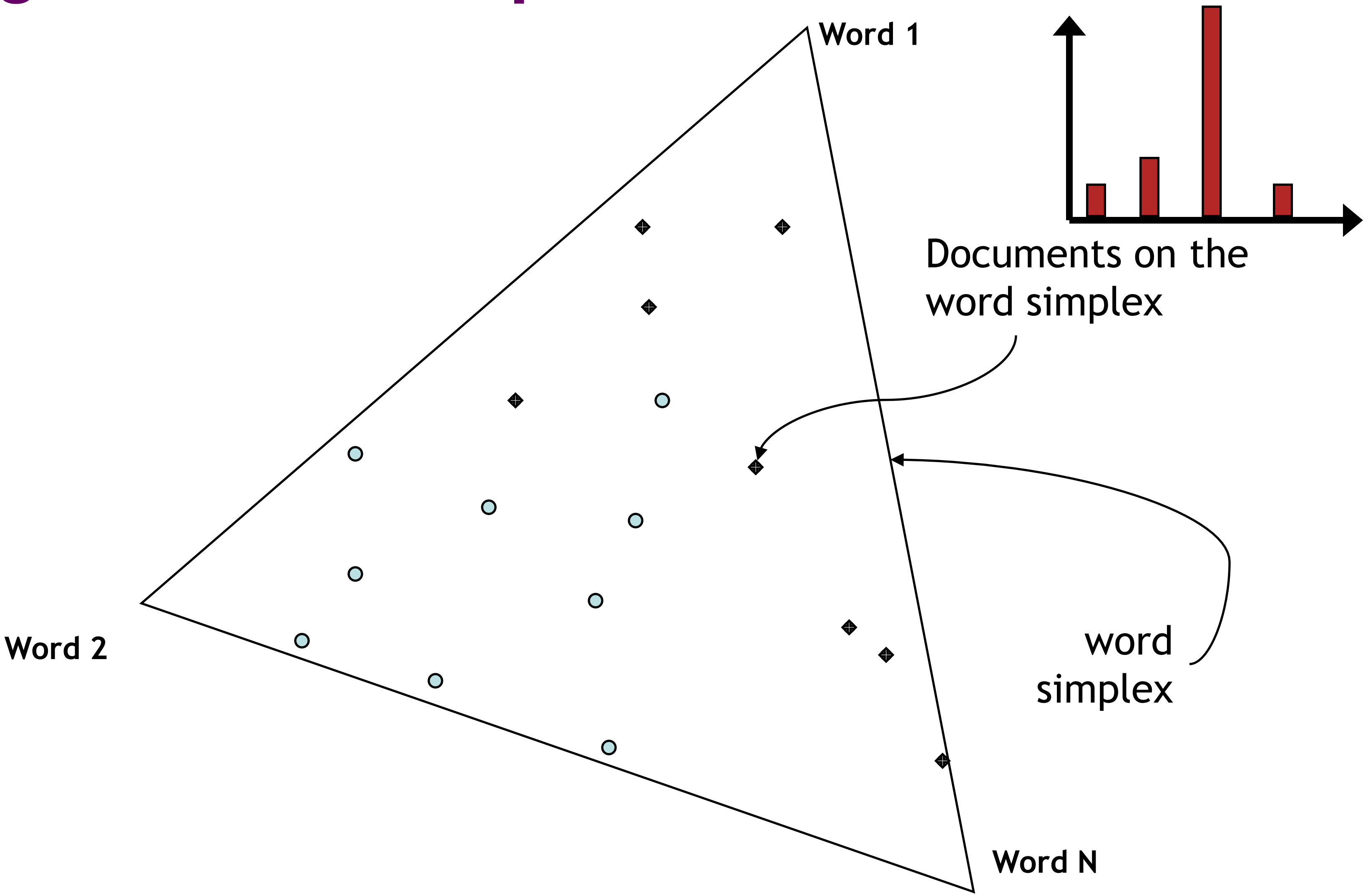


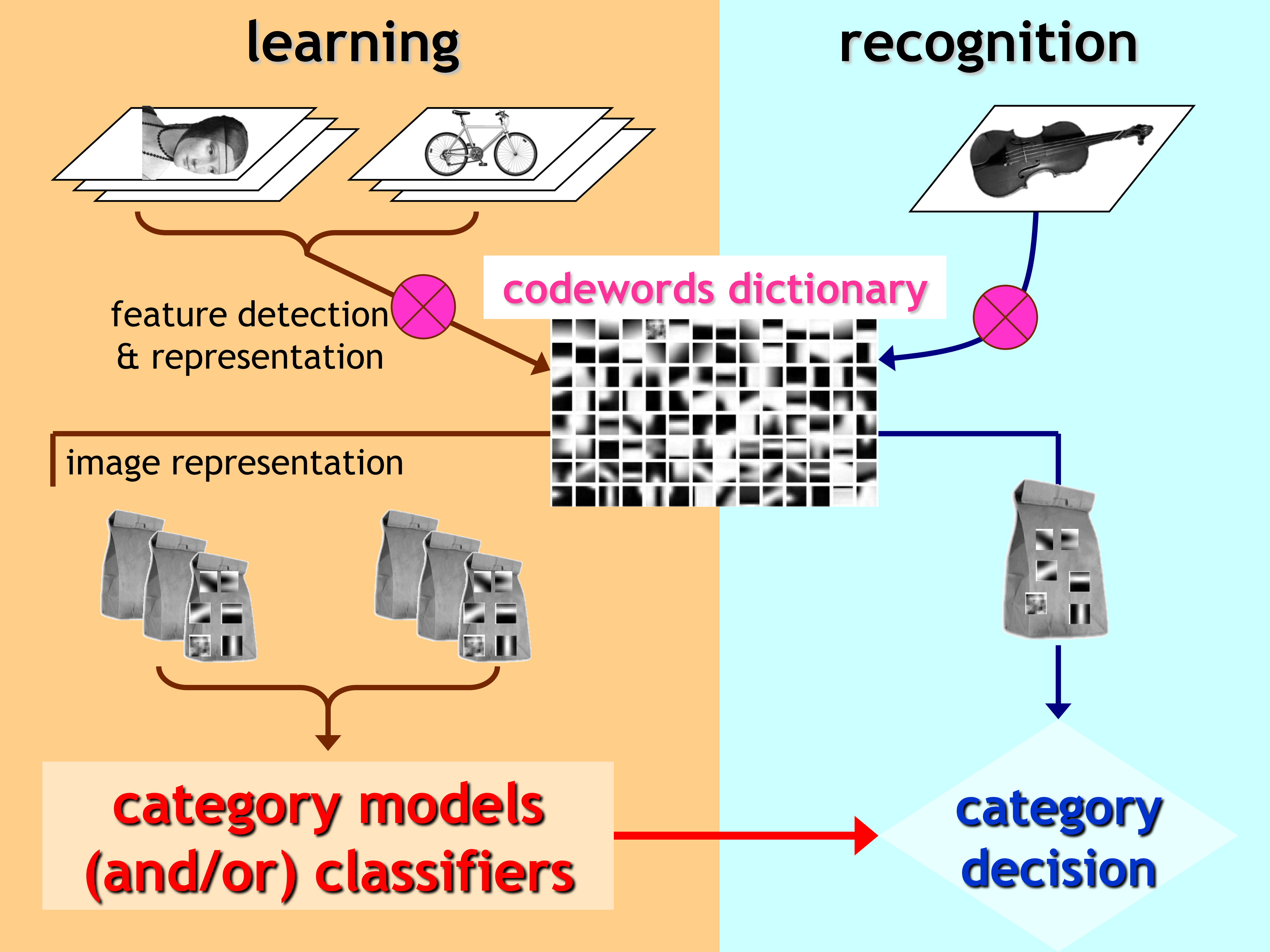
Revisit Bag-of-words

- An **image** ⇔ document is a collection of M **visual words** ⇔ words
- A corpus (collection of documents) is summarized in a term-document matrix



A geometric interpretation



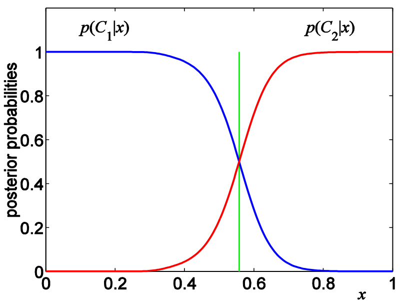

learning
recognition
feature detection
& representation
codewords dictionary
image representation
category models
(and/or) classifiers
category
decision

Modeling

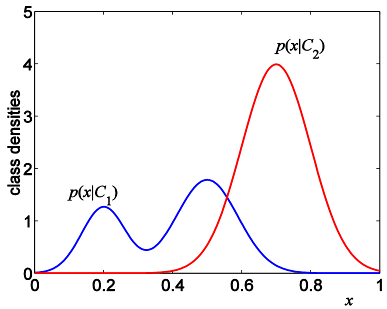
- So we want to look at
  - **high-dimensional visual data**, and
  - **fit models** to it;
  - **forming summaries** of it that let us understand what we see.

Learning and Recognition

1. Discriminative method:
   - SVM, logistic regression



2. Generative method:
   - graphical models



category models (and/or) classifiers

Probability Basics

- X is a random variable, P(X) is the probability that X achieves a certain value.
- Joint probability $$p(x, y)$$
- Conditional probability $$p(x|y)$$
- Sum Rule $$p(\mathbf{x}) = \sum_{\mathbf{y}} p(\mathbf{x}, \mathbf{y})$$
- Product Rule $$p(\mathbf{x}, \mathbf{y}) = p(\mathbf{x}|\mathbf{y})p(\mathbf{y})$$
- Bayes Rule $$p(\mathbf{y}|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{y})p(\mathbf{y})}{p(\mathbf{x})}$$

Generative Classifiers

- Learn a model of joint probability ***p(x,y)***
  - Of the inputs ***x*** and the label ***y***

Class conditional probability

$$p(\mathbf{x},\mathbf{y}) = p(\mathbf{x}|\mathbf{y})p(\mathbf{y})$$

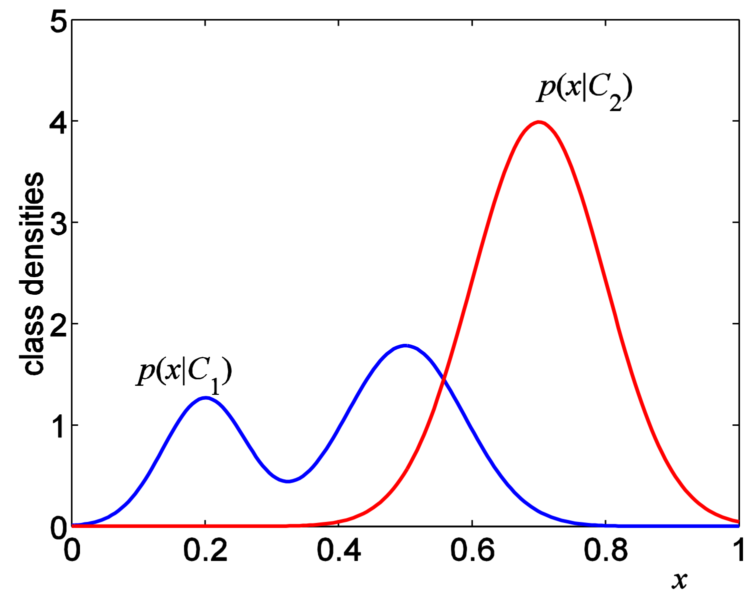
Class probability

- Make predictions by using Bayes Rule
  - To calculate ***p(y|x)*** and then picking most likely ***y***

$$p(\mathbf{y}|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{y})p(\mathbf{y})}{p(\mathbf{x})}$$

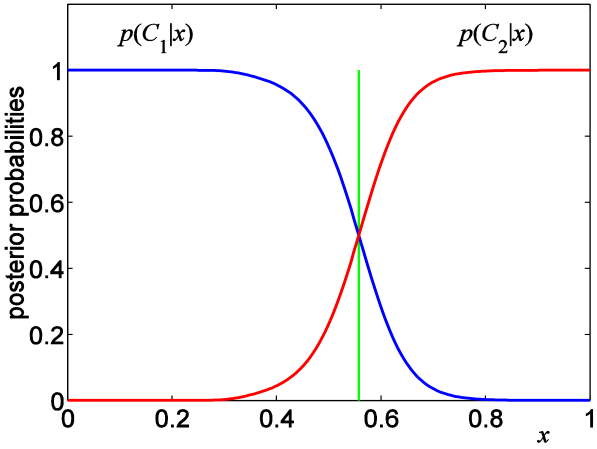
  - where

$$p(\mathbf{x}) = \sum_{\mathbf{y}} p(\mathbf{x}|\mathbf{y})p(\mathbf{y})$$



Discriminative Classifiers

- Model the posterior ***p(y|x)*** directly
  - Or learn a direct map from inputs x to the class labels



- Why discriminative?
  - "One should solve the [classification] problem directly and **never solve a more general problem** as an intermediate step [such as modeling ***p(x|y)***]." --- Vapnik, 1995
  - "Indeed leaving aside computational issues and matters such as handling missing data, the **prevailing consensus seems to be that discriminative classifiers** are almost **preferred** to generative ones" --- Ng. and Jordan, 2002

What is the difference asymptotically?

Notation: let $\epsilon(h_{A,m})$ denote error of hypothesis learned via algorithm A, from *m* examples

- If **assumed model correct** (e.g., naïve Bayes model), and **finite number of parameters**, then

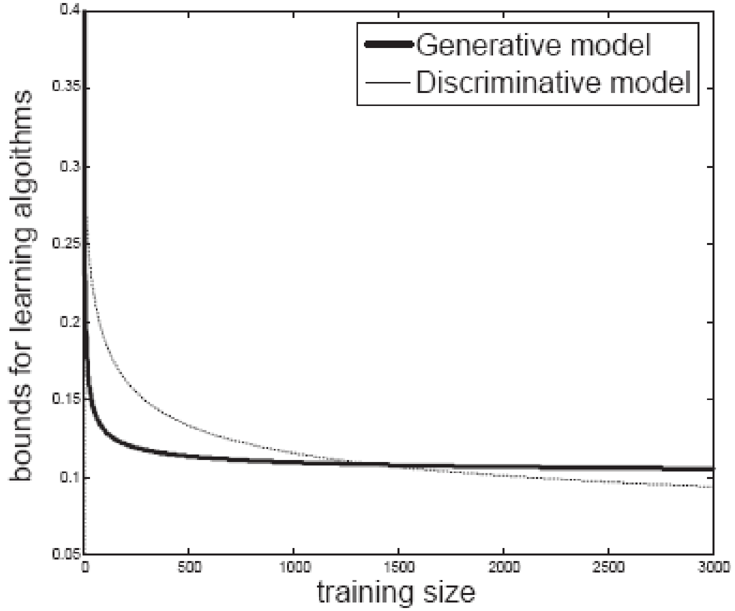
$$\epsilon(h_{Dis,\infty}) = \epsilon(h_{Gen,\infty})$$

- If assumed **model incorrect**

$$\epsilon(h_{Dis,\infty}) \leq \epsilon(h_{Gen,\infty})$$

Then why Generative Models?

- Although generative models have higher asymptotic error
  - The generative model may approach its asymptotic error much faster than discriminative model
  - i.e. when the number of training examples are less, generative models might out perform discriminative models

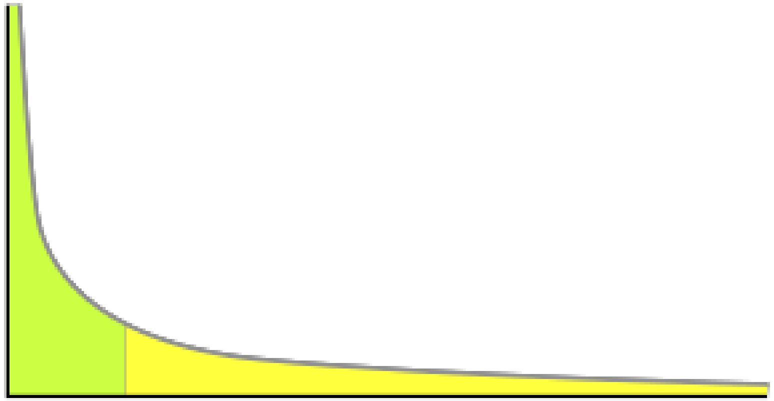


  - *Andrew Y. Ng , Michael I. Jordan, "On Discriminative vs. Generative classifiers: A comparison of logistic regression and naive Bayes", NIPS (2002)*

Recently…

- Rubin et. al. showed that "*on large-scale datasets with power-law like statistics, the Dependency-LDA [a generative model] models generally outperforms binary SVMs [a discriminative model]*"
  - Statistical Topic Models for Multi-Label Document Classification, Nov 2011

- Power-law statistics
  - Several classes have very few training instances



Plus…

- Discriminative models have
  - **Limited modeling capability**. Can not generate new data.
  - Require both positive and negative training data (mostly).
  - **Still, are the first choice for classification tasks**.

- Many approaches that combine discriminative and generative models
  - Raina, Rajat, et al. "Classification with hybrid generative/discriminative models." *Advances in neural information processing systems* 16 (2003).
  - Lasserre, Julia A., Christopher M. Bishop, and Thomas P. Minka. "Principled hybrids of generative and discriminative models." *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*. Vol. 1. IEEE, 200

Generative Classifiers

- Learn a model of joint probability ***p(x,y)***
  - Of the inputs ***x*** and the label ***y***

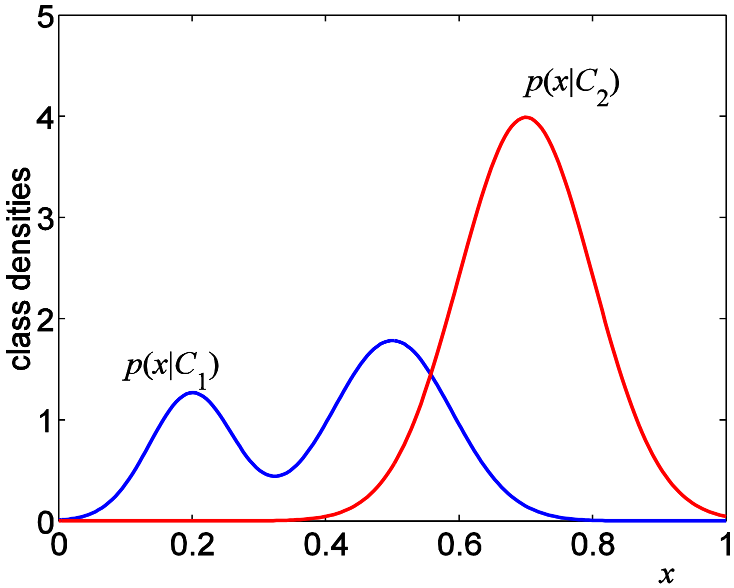
$$p(\mathbf{x}, \mathbf{y}) = p(\mathbf{x}|\mathbf{y})p(\mathbf{y})$$

- Make predictions by using Bayes Rule
  - To calculate ***p(y|x)*** and then picking most likely ***y***

$$p(\mathbf{y}|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{y})p(\mathbf{y})}{p(\mathbf{x})}$$

  - where

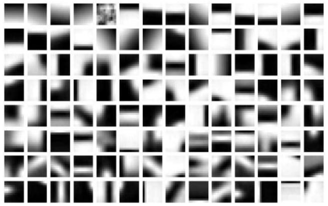
$$p(\mathbf{x}) = \sum_{\mathbf{y}} p(\mathbf{x}|\mathbf{y})p(\mathbf{y})$$



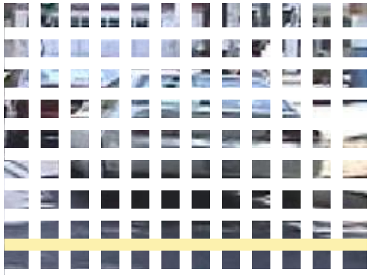
Case Studies

- Naïve Bayes classifier
  - Csurka et al. 2004
- Hierarchical Bayesian models (pLSA and LDA)
  - Hoffman 2001,
  - Blei et al. 2004
- Supervised Hierarchical Bayesian models
  - Supervised LDA: Blei et al. 2007
  - Natural scene categorization: Fei-Fei et al. 2005
  - Class-Specific-Simplex LDA: Rasiwasia et al. 2012
  - DiscLDA, MedLDA, Semi-LDA, etc.

First, some notations



- Codebook $\mathcal{V} = \{v_1, \ldots, v_L\}$ with L codewords
- $\mathbf{w} = \{w_1, \ldots, w_N\}$ collection of N patches in an image,

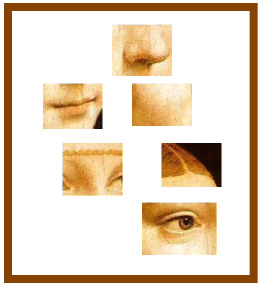


- where each patch, $w_n \in \mathcal{V}$
- $\mathcal{D} = \{(\mathbf{w}_1, c_1), \ldots, (\mathbf{w}_M, c_M)\}$ dataset of M images,
- where each image is labeled with category c_m

Naïve Bayes Model

- Assume that each feature is conditionally independent *given the class*

$$p(w_1, \ldots, w_N \mid c) = \prod_{n=1}^{N} p(w_n \mid c)$$

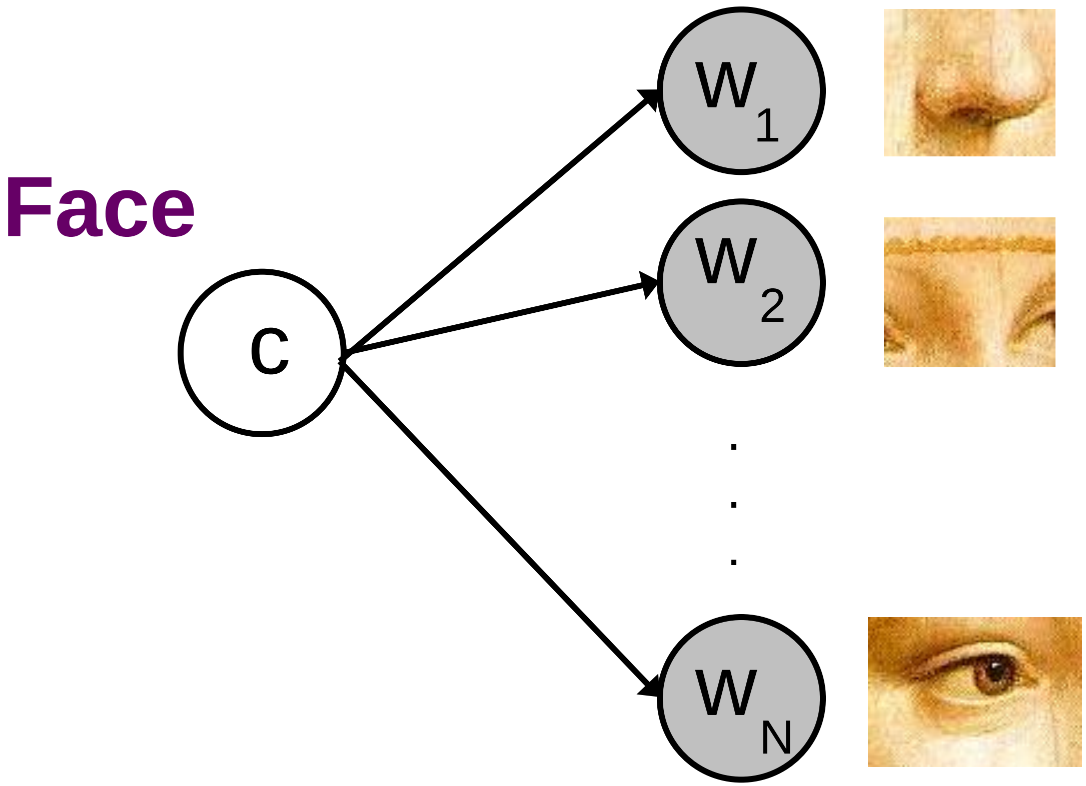


w_n: nth feature / patch in the image
N: number of features in the image

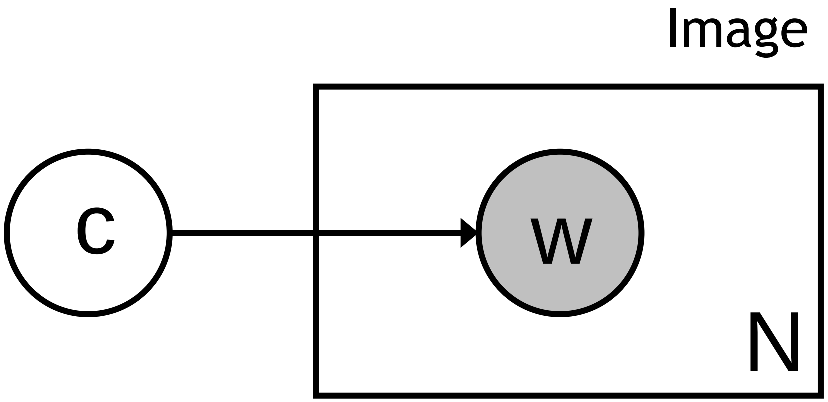
Naïve Bayes

- Representation using Probabilistic Graphical Models
- Tool for representing complex systems and performing sophisticated reasoning tasks
- Fundamental notion: Modularity
  - Complex systems are built by combining simpler parts

Graphical Model

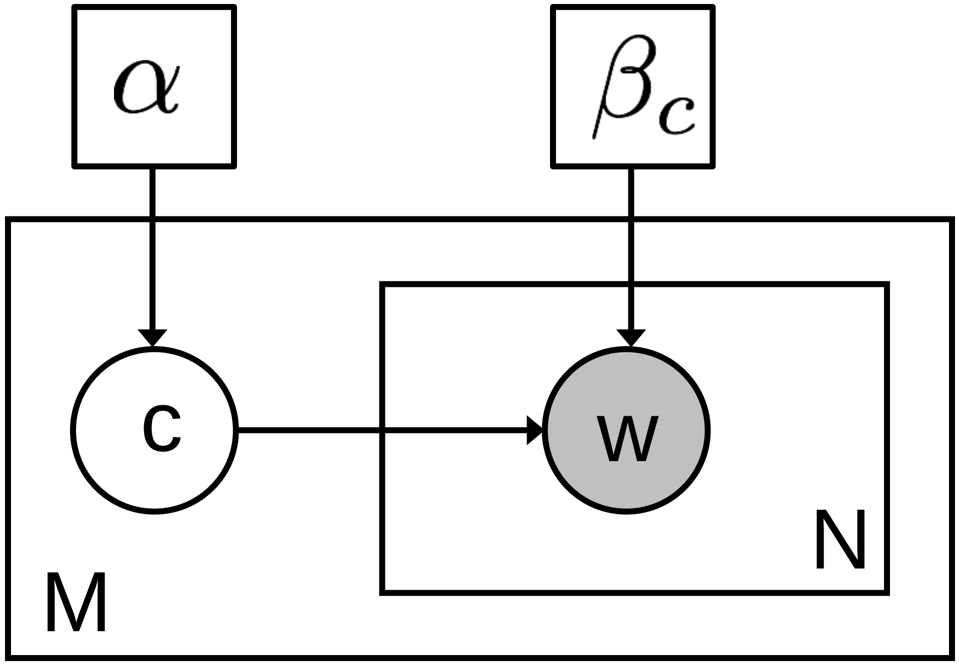


Graphical Model



Graphical Model

Parameters of the model



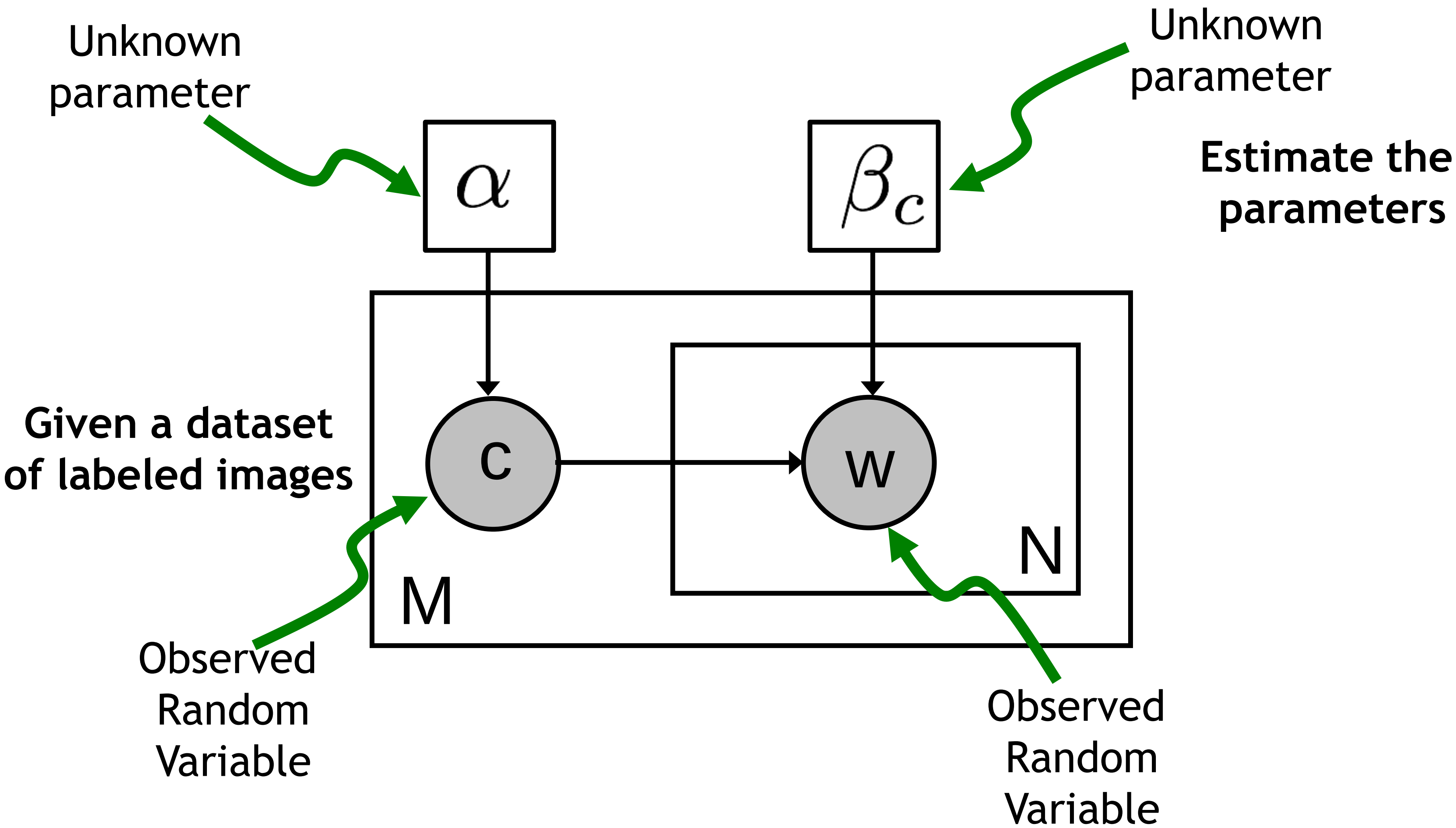
Collection of Images

$$P(c = j; \alpha) = \alpha_j \qquad P(w_n = v_j | c; \beta_c) = \beta_{cj}$$

Prior probability **class conditional distribution**

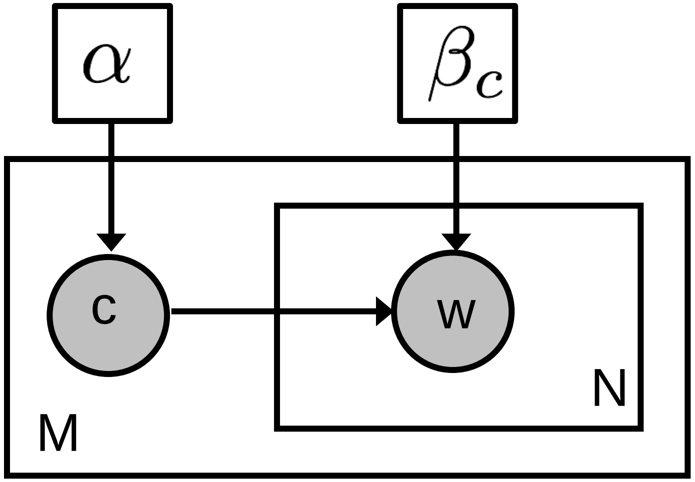
$$P(c; \alpha) = \prod_{j=1}^{C} \alpha_j^{\delta(c,j)} \qquad P(w_n | c; \beta_c) = \prod_{j=1}^{L} \beta_{cj}^{\delta(w_n, v_j)}$$

Graphical Model - Parameter Learning



Learning: Maximum likelihood estimation

Maximize (log)likelihood of data:



$$L = \prod_{m=1}^{M} p(\mathbf{w}_m, c_m)$$

$$= \prod_{m=1}^{M} p(c_m) \prod_{n=1}^{N} p(w_{mn}|c_m)$$

$$\log L = \sum_{m=1}^{M} \log p(c_m) + \sum_{m=1}^{M} \sum_{n=1}^{N} \log p(w_{mn}|c_m)$$

$$\log L = \sum_{m=1}^{M} \sum_{l=1}^{C} \delta(c_m, l) \log \alpha_l + \sum_{m=1}^{M} \sum_{n=1}^{N} \sum_{j=1}^{L} \delta(w_{mn}, j) \log \beta_{cj}$$

$$(\alpha, \beta)^* = \arg\max_{\alpha,\beta} \log L$$

Learning: Maximum likelihood estimation

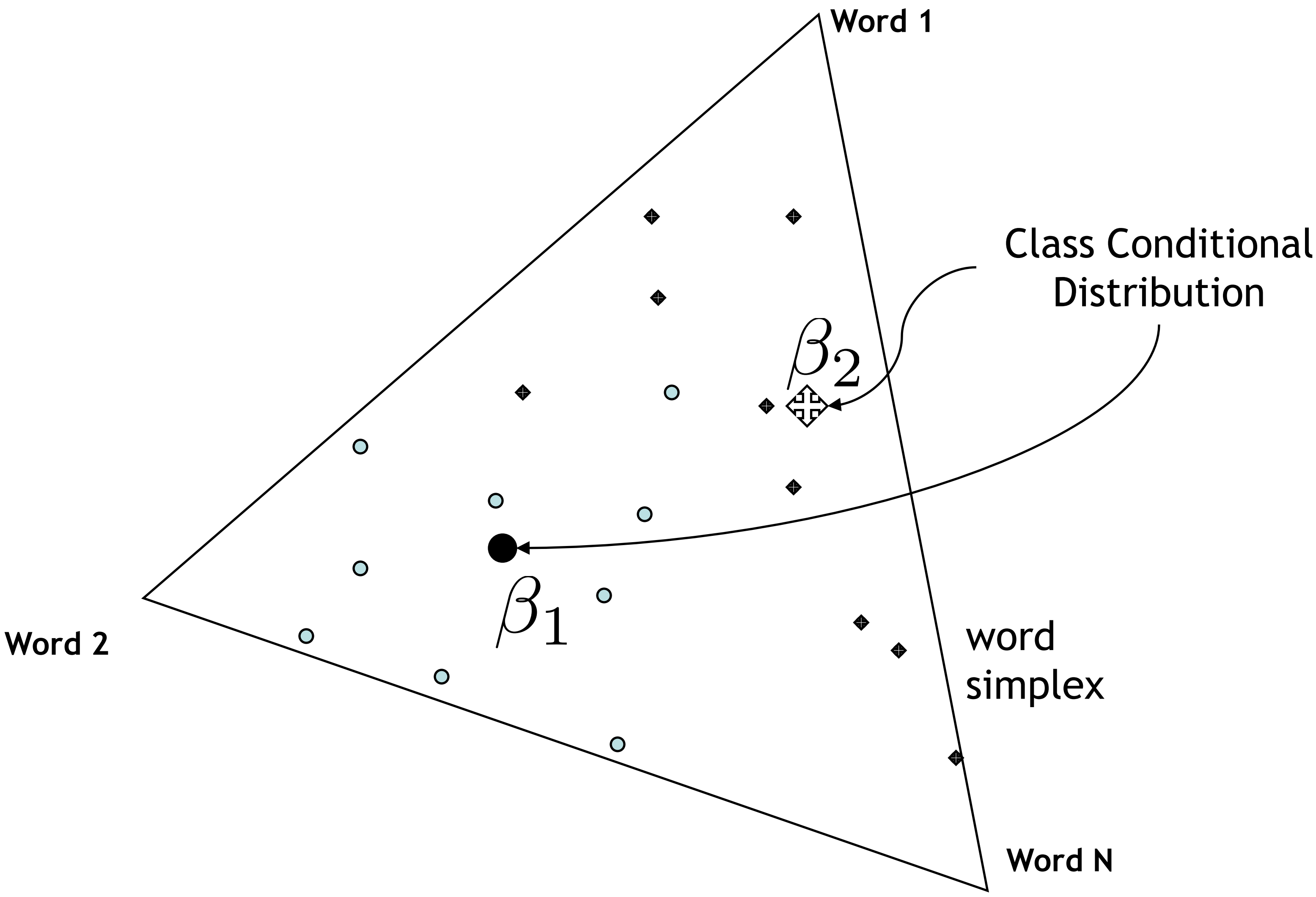
- ***Using standard Lagrangian formulation***,
- Class conditional distribution,

$$\beta_{cj} = \frac{\text{No. of features of codeword } v_j \text{ in training images of class } c}{\text{Total no. of features in training images of class } c}$$

- and the prior distribution,

$$\alpha_l = \frac{\text{No. of training images of class } l}{\text{Total no. of images}}$$

A geometric interpretation



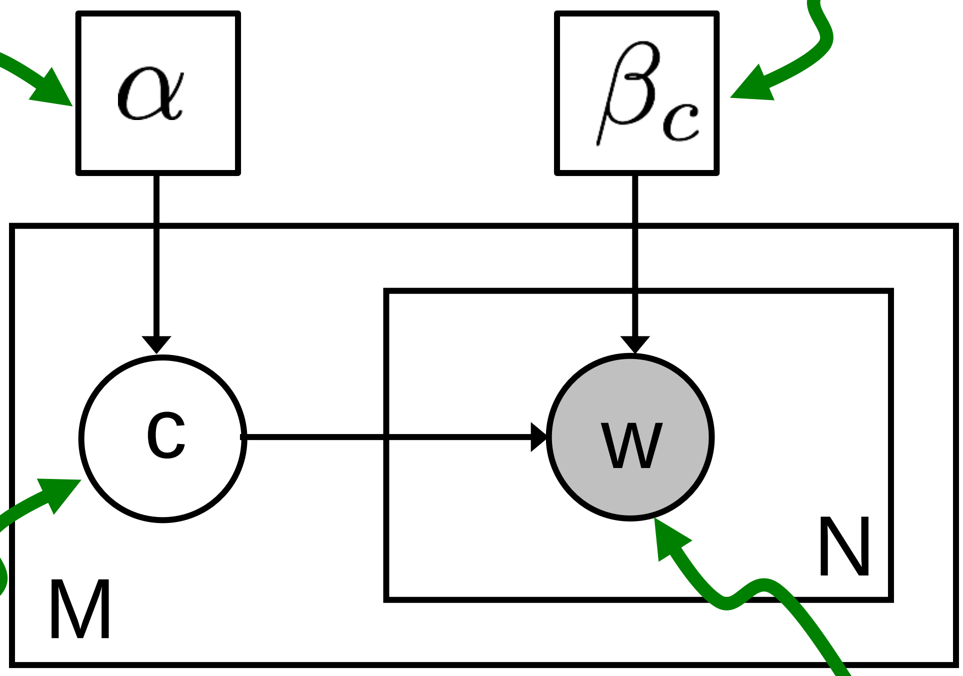
Graphical Model - Inference

Learned parameter

Learned parameter

And the learned parameters

Infer the class



Given a new image

Unobserved Random Variable to be inferred

Observed Random Variable

Csurka et al. 2004

Generative Models

During recognition, **minimum probability of error classification** rule says, classify the image (w) to the class of highest posterior probability,

$$\begin{aligned} c^* &= \arg\max_c p(c|\mathbf{w}) \\ &= \arg\max_c \log p(c|\mathbf{w}) \end{aligned}$$

Inference

Prior prob. of
the image classes

$$p(c|\mathbf{w};\alpha,\beta_c) \propto p(c)p(\mathbf{w}|c)$$

$$= p(c)\prod_n p(w_n|c)$$

Image likelihood
given the class

Class posterior

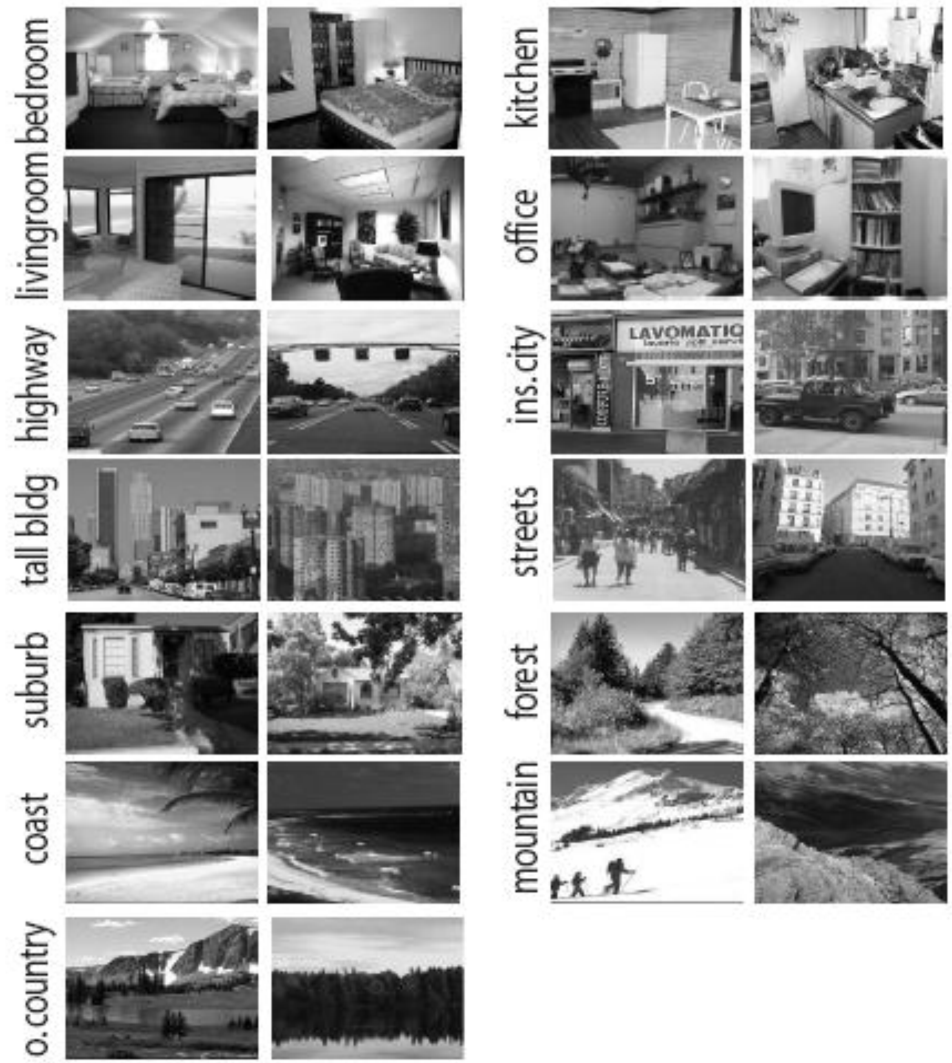
$$= \prod_k \alpha_k^{\delta(c,k)} \prod_n \prod_j \beta_{cj}^{\delta(w_n,j)}$$

$$= \prod_k \alpha_k^{\delta(c,k)} \prod_j \beta_{cj}^{n_j}$$

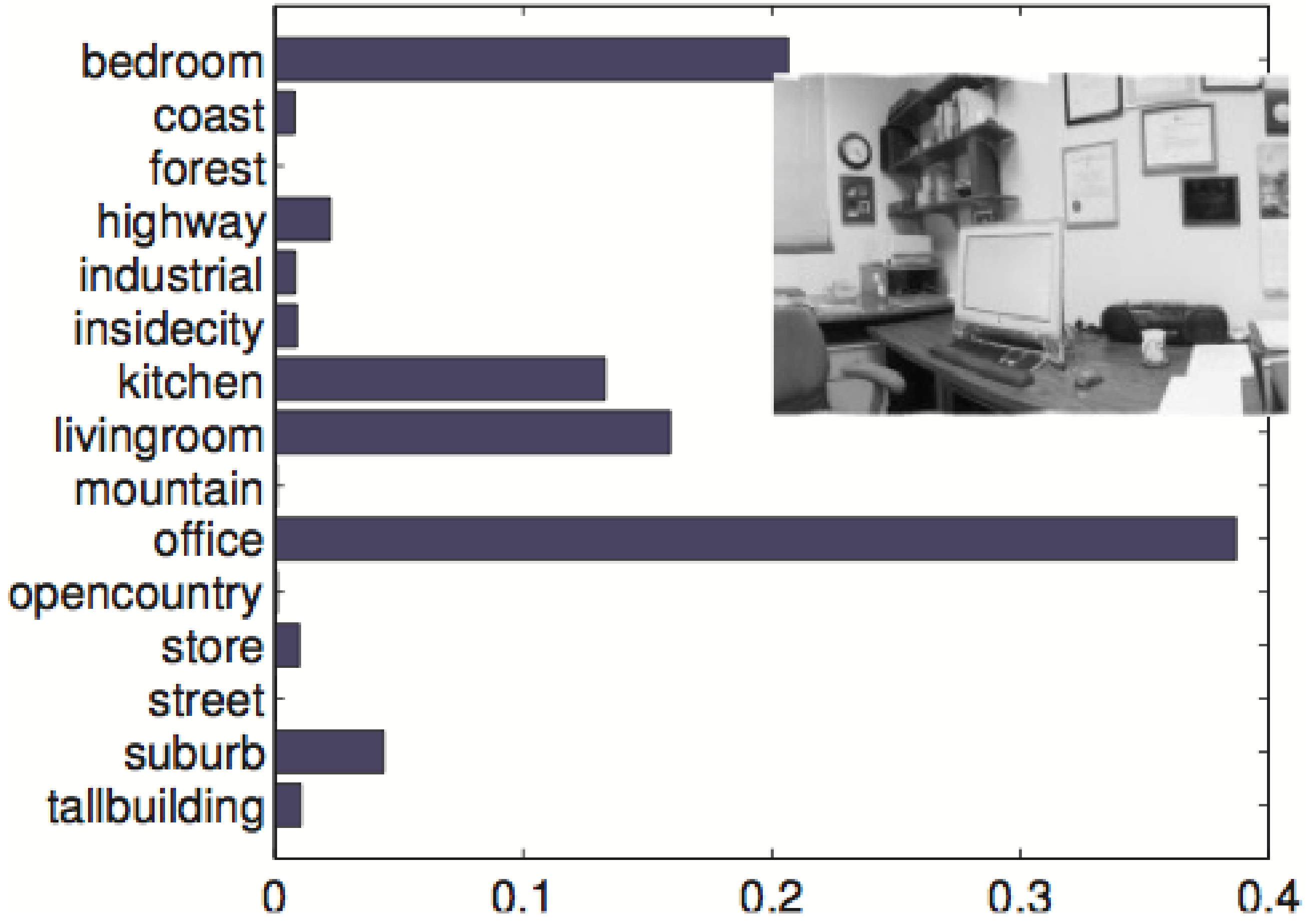
$$\log p(c|\mathbf{w};\alpha,\beta_c) = \sum_k \delta(c,k)\log\alpha_k + \sum_j \boxed{n_j}\log\beta_{cj}$$

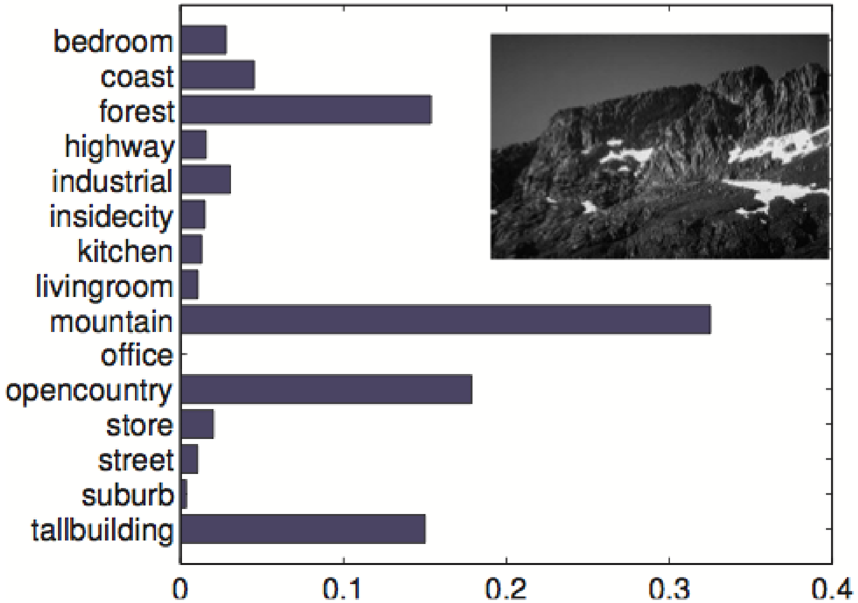
Evaluation: Scene categorization

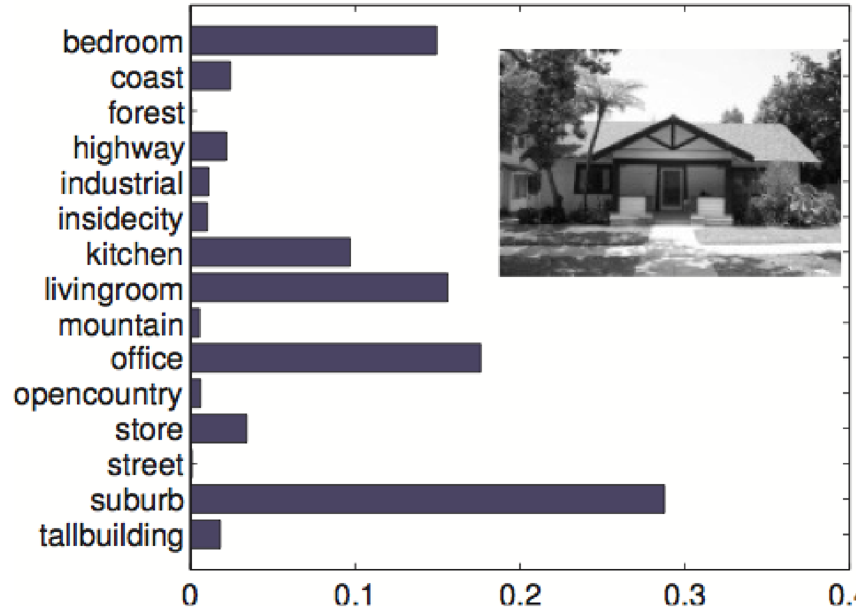
- Dataset
  - 15 scene categories
- Learning
  - 100 images per class
- Testing
  - ~200 images per class
- Codebook Size
  - 4000 codewords
- Classification Rate
  - 74.91 % correct classification



Some Examples







→ Kitchen
Bedroom (36%)

→ Office
Livingroom (55%)

→ Inside City
Street (66%)

→ Store
Kitchen (66%)

Confusion Matrix

- Confusion between
  - Office, Bedroom Livingroom Kitchen
  - Store, Industrial
  - Highway, Coast Opencountry

	Office	Livingroom	Bedroom	Kitchen	Store	Industrial	TallBuilding	InsideCity	Street	Highway	Coast	Opencountry	Mountain	Forest	Suburb
Office	.96	.01	.02	.01	.00	.00	.01	.00	.00	.00	.00	.00	.00	.00	.00
Livingroom	.05	.55	.19	.05	.01	.00	.02	.03	.08	.00	.01	.01	.00	.00	.01
Bedroom	.09	.25	.36	.14	.03	.03	.01	.02	.04	.01	.00	.00	.02	.00	.00
Kitchen	.07	.07	.04	.66	.08	.05	.00	.02	.01	.00	.00	.00	.00	.00	.00
Store	.00	.02	.01	.07	.80	.08	.00	.00	.01	.00	.00	.00	.00	.00	.00
Industrial	.00	.00	.05	.04	.12	.69	.03	.01	.02	.00	.00	.01	.00	.01	.00
TallBuilding	.00	.02	.02	.00	.01	.04	.71	.05	.09	.02	.00	.00	.00	.01	.03
InsideCity	.00	.03	.01	.02	.01	.02	.06	.74	.07	.01	.00	.00	.00	.00	.01
Street	.00	.03	.03	.01	.02	.02	.10	.06	.66	.06	.00	.01	.00	.00	.02
Highway	.00	.01	.01	.00	.01	.02	.01	.02	.02	.78	.05	.06	.02	.00	.00
Coast	.00	.00	.00	.00	.00	.02	.00	.01	.00	.04	.77	.13	.02	.00	.00
Opencountry	.00	.00	.01	.00	.01	.00	.00	.00	.01	.04	.10	.66	.08	.08	.00
Mountain	.01	.00	.01	.00	.01	.01	.01	.00	.01	.01	.04	.10	.71	.07	.01
Forest	.00	.00	.00	.00	.04	.00	.01	.00	.01	.01	.00	.05	.06	.82	.01
Suburb	.00	.00	.00	.00	.00	.00	.02	.01	.00	.00	.00	.00	.00	.01	.96